

افزایش کیفیت ویدیو دریافتی با استفاده از کدگذاری ثانویه کانال در کدگذاری همزمان منبع و کانال

پوریا اعتضادی فر^۱، حسن فرسی^{۲*}، علی ناصری^۳

۱- دانشجوی دکتری دانشگاه بیرجند، ۲- دانشیار، دانشگاه بیرجند، ۳- دانشیار، دانشگاه جامع امام حسین(ع)
(دریافت: ۹۳/۱۰/۱۶؛ پذیرش: ۹۴/۰۶/۱۰)

چکیده

با توجه به پیشرفت تکنولوژی در دهه‌های اخیر، ارسال و دریافت ویدیو از طریق کانال‌های مخابراتی بی‌سیم با استقبال گسترده‌ای روبرو شده است. بدین منظور روش‌های بسیاری جهت بالا بردن کیفیت ویدیویی ارسالی ارائه گردیده‌اند. در این مقاله هدف بالا بردن PSNR (حداکثر نرخ توان سیگنال به توان نویز) به منظور بالا بردن کیفیت فریم‌های ویدیویی دریافتی در گیرنده با استفاده از بالا بردن نرخ کدکننده کانال در یک نرخ ارسال ثابت می‌باشد. اساس کار بدین صورت است که با استفاده از شبکه عصبی هوشمند و کدها فم در استاندارد MPEG (نوعی فشرده‌ساز ویدیویی استاندارد شده)، اطلاعات ارسالی به مقدار قابل توجهی فشرده می‌گردد؛ سپس با توجه به مقدار فشرده‌سازی توسط روش پیشنهادی، اطلاعات فشرده شده را مجدداً با کدکننده کانال ثانویه که نرخ کدگذاری آن وابسته مقدار فشرده‌سازی می‌باشد کم می‌شود. به این ترتیب روش پیشنهادی قادر است بدون افزایش حجم اطلاعات ارسالی برای هر فریم، نرخ کدگذاری کانال و در نتیجه محافظت از اطلاعات را در مقابل افزایش اعوجاج و خطای کانال بالا برده و توانسته است فریم‌های ویدیویی را نسبت به خطاهای کانال مقاوم‌تر سازد. در نهایت نتایج به دست آمده را با چندین نرخ ارسال برای منبع و چندین SNR (نرخ توان سیگنال به توان نویز) برای کانال با نتایج به دست آمده از روش‌های متداول مقایسه می‌گردد.

واژه‌های کلیدی: کدگذاری فریم‌های ویدیویی، نرخ بیت متغیر، کدگذاری کانال، تطبیق کدگذاری کانال و منبع.

روش‌های متداولی برای کدگذاری همزمان منبع و کانال وجود دارد که عموماً بر اساس تخمین کانال صورت می‌پذیرد [۳-۶]. روش به کار گرفته شده در این مقاله با روش‌های اشاره شده در مراجع [۴-۶] متفاوت می‌باشد. اساس این روش به منظور بالا بردن نرخ کدکننده کانال می‌باشد. به این معنی که در یک نرخ ارسال ثابت بتوان نرخ کدکننده کانال را افزایش داد. با افزایش نرخ کدکننده کانال، فریم‌های ویدیویی ارسالی را می‌توان نسبت به کانال مقاوم‌تر ساخت. یکی از مشکلات روش‌های به کار گرفته شده در مراجع [۲-۵] این است که باید از کانال با خبر باشند و بسته به اعوجاج کانال نرخ کدکننده منبع و کانال را تغییر دهند. اما روش پیشنهادی وابسته به کانال نبوده و قادر است بر روی هر نوع کدگذاری منبعی به صورت مستقل عمل نموده و کیفیت تصاویر دریافتی در گیرنده را افزایش دهد. در ادامه ابتدا به معرفی چند پارامتر استفاده شده در مقاله پرداخته می‌شود و در قسمت بعد به توضیح پیکره‌بندی مقاله اشاره‌ای خواهیم داشت. در این مقاله اساس سنجش فریم‌ها بر حسب

۱- مقدمه

رشد روزافزون استفاده از مخابرات بی‌سیم و تمایل به استفاده از دستاوردهای مخابرات مدرن برای ارسال داده‌های چندرسانه‌ای^۱ بی‌درنگ همچون ویدیو، ضرورت تحقیق و مطالعه در این زمینه را نمایان می‌کند. روش‌های مختلف کدگذاری ویدیو برای رسیدن به نرخ بیت پایین‌تر و در عین حال با کاهش کیفیت اندک مطرح گردیده‌اند [۱]. در مخابرات بی‌سیم و سیار خطای ذاتی محیط‌های بی‌سیم منجر به کاهش کیفیت ویدیو دریافتی در گیرنده می‌شود. تاکنون روش‌های بسیاری برای کاهش و حذف خطای ناشی از ارسال ویدیو در محیط‌های بی‌سیم ارائه شده است [۲]. یکی از کارآمدترین این روش‌ها کدگذاری کانال است که برای تشخیص و تصحیح خطا به کار گرفته می‌شود. در سال‌های اخیر، روش‌های متنوعی در رابطه ترکیب توأم کدگذاری منبع و کانال^۲ معرفی گردیده است.

* رایانامه نویسنده مسئول: hfarsi@birjand.ac.ir

1- Multimedia

2- Joint Source and Channel Coding (JSCC)

سازی این تصاویر از روش‌های فشرده سازی مثل JPEG^۸ استفاده می‌شود [۹-۷]. به تصویری که با استفاده از نزدیکترین تصاویر P و I قبلی تخمین زده می‌شوند، تصویر P گفته می‌شود. در این نوع کدگذاری از جبران ساز حرکت نیز استفاده می‌گردد. به تصویری که با استفاده از تصاویر P و I قبلی و بعدی که به‌عنوان مرجع هستند تخمین زده می‌شوند، تصویر B گفته می‌شود. در این نوع کدگذاری نیز از جبران ساز حرکت استفاده می‌گردد. در این مقاله تنها دو نوع فریم تصویر I و تصویر P را از گروه تصاویر انتخاب نموده‌ایم. علت انتخاب این تصاویر آن است که در تصویر B به دلیل این که تخمین با استفاده از فریم‌های قبل و بعد صورت می‌پذیرد، سیستم باید تا زمانی که فریم بعد وارد و ذخیره گردد منتظر بماند تا بتواند با استفاده از آن تخمین انجام دهد. بنابراین سیستم دچار تاخیر می‌گردد. از این رو از تصویر B در این نوع کدگذاری استفاده نگردیده است.

۲-۱- مروری بر روش کد کردن MPEG-4

در این مقاله از MPEG-4 part 2 استفاده نموده‌ایم [۷]. ساختار کدکننده و کدبرگردان I و P به ترتیب در شکل‌های (۲) و (۳) نشان داده شده است. در ادامه به توضیح برخی از عملیات استفاده شده در انکدر و دیکدر نشان داده شده در شکل‌های (۲) و (۳) (البته به غیر از قسمت کدکننده کانال) می‌پردازیم.

۲-۱-۱- مرتب‌سازی مجدد^۹

بعد از کوانتیزه کردن ضرایب DCT، با توجه به زیاد بودن مقادیر صفر، ضرایب به صورت زیگزاگ کنار همدیگر قرار داده می‌شوند. با این کار مقادیر مربوط به فرکانس‌ها گروه‌بندی می‌شود [۱۰-۱۱]. نحوه چیدمان زیگزاگ در شکل (۴) و نحوه چیدمان زیگزاگ متناوب در شکل (۵) نشان داده شده است.

۲-۱-۲- بسته‌بندی کردن داده‌ها^{۱۰}

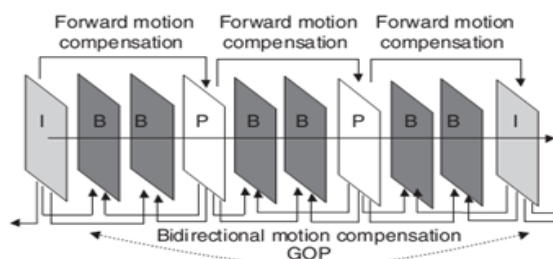
وجود خطا در هنگام ارسال داده‌ها باعث از بین رفتن همزمانی بین فریم‌های ارسالی می‌گردد. برای فائق آمدن به این مشکل نیاز به مکانیزم همزمان‌ساز مجدد می‌باشد. یکی از راه‌های همزمان‌سازی فریم‌های ارسالی استفاده از ارسال داده‌ها به صورت بسته‌بندی شده^{۱۱} می‌باشد. به این منظور در هر بسته تعدادی نشانه^{۱۲} برای همزمان‌سازی قرار داده می‌شود. کدکننده هر فریم را به بسته‌هایی تقسیم می‌کند و نشانه‌های مربوط به همزمانی را در ابتدای هر بسته قرار می‌دهد. کدکننده و کدبرگردان هنگام رسیدن به نشانه‌های قرارداده شده در اول هر

ماکزیم نسبت توان سیگنال به توان نویز PSNR^۱ می‌باشد و همچنین BER^۲ (نرخ خطای بیتی) است که میزان خطای ایجاد شده توسط کانال بر روی اطلاعات را بیان می‌کند. این مقاله به صورت زیر ساماندهی شده است:

در بخش ۲، ابتدا روش کد کردن فریم‌های ویدیویی با استفاده از MPEG-4 part 2 (که با شماره سند ISO/IEC 14496-2 به صورت بین‌المللی استاندارد گردیده است) برای دو فریم تصویر I^۳ (تصویر داخلی) و تصویر P^۴ (تصویر تخمین زده شده) را معرفی می‌نماییم. در بخش ۳، کدگذاری کانالی که در این مقاله استفاده شده است را بررسی می‌نماییم. در بخش ۴، نحوه به دست آوردن تابع چگالی احتمال برای یک رشته با طول دلخواه را بررسی خواهیم نمود و پس از آن در بخش ۵، به مفهوم شبکه عصبی اشاره‌ای خواهیم داشت. در بخش ۶، به مفهوم حافظه‌دار کردن رشته اطلاعات و سپس به مزیت حافظه‌دار کردن رشته‌ها پرداخته می‌گردد. در بخش ۷، روش پیشنهادی را مطرح نموده و به صورت کامل به توضیح آن می‌پردازیم، سپس در بخش ۸، به ارزیابی روش پیشنهادی و مقایسه داده‌های خروجی با روش به کار گرفته شده در چندین مرجع پرداخته می‌شود.

۲- نحوه کدگذاری فریم‌های ویدیویی

در این بخش، ابتدا به صورت خلاصه به معرفی گروه تصاویر^۵ می‌پردازیم و در مرحله بعد نحوه کد و دیکد کردن فریم‌های ویدیویی با استفاده از MPEG-4 را توضیح می‌دهیم. در استاندارد MPEG، ۳ نوع تصویر معرفی می‌گردند که به اختصار با I و P و B^۶ (تصویر تخمین زده شده دوجهته) نشان داده می‌شوند. از ترکیب این تصاویر با هم گروه تصاویر ساخته می‌شود. نمونه‌ای از ساختار گروه تصاویر در شکل (۱) نشان داده شده است.



شکل (۱). نمونه‌ای از ساختار GOP [۷]

به تصویری که با استفاده از اطلاعات موجود در خود آن تصویر کدی می‌شوند، تصویر I گفته می‌شود. بنابراین برای فشرده

8- Joint Photographic Experts Group
9- Reorder
10- Packetization
11- Packetize
12- Marker

1- Peak Signal to Noise Ratio
2- Bit Error Rate
3- Intra-picture
4- Predicted-Picture
5- Group Of Pictures (GOP)
6- Bi-directional Predicted Frames
7- Joint Photographic Experts Group

۲-۱-۴- بلوک تخمین گر حرکت

بلوک تخمین گر حرکت بر اساس معیار محاسبه کمترین میانگین مربعات خطای تطبیق بین دو فریم متوالی کار می کند که نحوه کار آن بر اساس پیاده سازی معادلات (۱ و ۲) می باشد.

$$MSE(d_x, d_y) = \frac{1}{M_1 N_1} \sum_{(m,n \in W)} (b[m, n, k] - b[m - d_x, n - d_y, k - 1])^2 \quad (1)$$

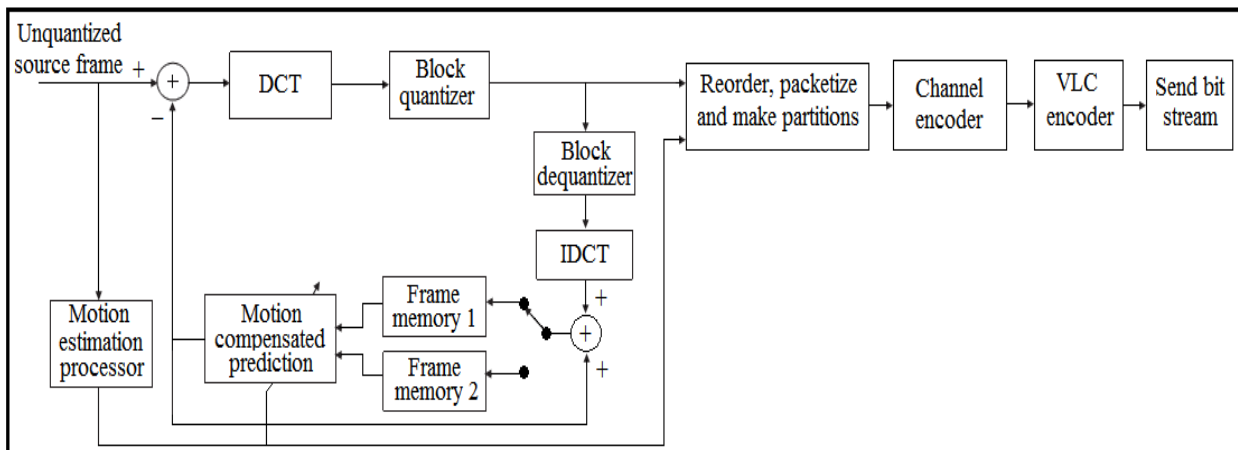
$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \arg \min_{(d_x, d_y)} MSE(d_x, d_y) \quad (2)$$

در معادله (۱) آرگمان سوم متغیر b شامل k و $k-1$ پیکسل های دو فریم پشت سرهم را فراخوانی می کند و دو متغیر m و n مکان یک پیکسل را در فضای دو بعدی هر فریم نشان می دهد. متغیر W نشان دهنده پنجره ای است که باید در آن بازه جستجو صورت گیرد و بردار تخمین حرکت با جستجو در این بازه به دست می آید که در این مقاله $W=15$ در نظر گرفته شده است.

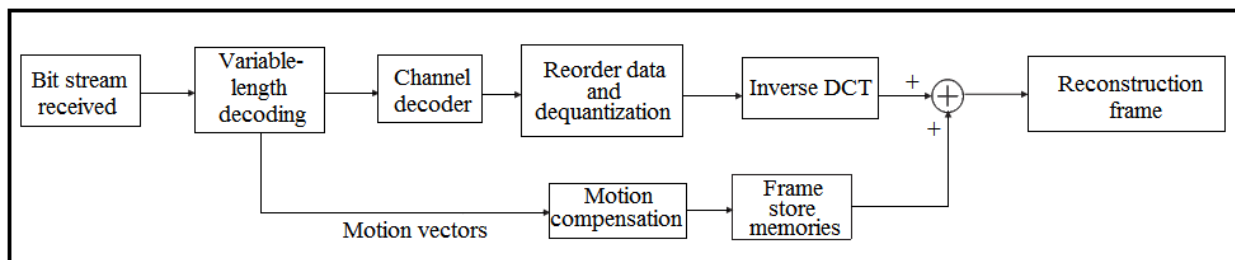
بسته، عملیات سنکرون سازی را مجددا انجام می دهند. به این ترتیب باعث می شوند که همزمان سازی بین فریم ها از بین نرود و از ایجاد خطای جمع شوندگی جلوگیری شود [۱۲].

۲-۱-۳- قسمت بندی کردن داده ها^۲

در این مدل داده های بسته بندی شده به دو ناحیه تقسیم می شوند. ایده اصلی این مدل، جداسازی داده ها با ارزش بیشتر (ضرایب DC ماتریس DCT، اطلاعات روش کدگذاری و بردارهای حرکت) از داده ها با ارزش پایین تر (ضرایب AC ماتریس DCT و خطاهای باقیمانده) می باشد. برای ارسال تصویر I اولین ناحیه، شامل اطلاعات روش کدگذاری منبع و همچنین ضرایب DC است. اما ناحیه دوم که داده های با ارزش پایین تر را داراست، شامل ضرایب AC می باشد. برای ارسال تصویر P اولین ناحیه شامل اطلاعات روش کدگذاری منبع و همچنین بردارهای حرکت می باشد، در حالی که ناحیه دوم شامل اطلاعات DCT (باقت، ضرایب DC و AC) می باشد. برای همزمان سازی در قسمت بندی کننده داده ها نیز از نشانه گذاری استفاده می شود که این نشانه گذاری یکتا می باشد.

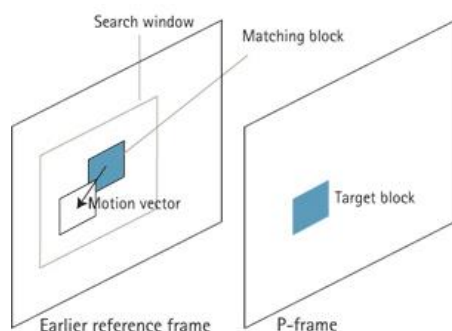


شکل (۲). کدکننده MPEG-4 برای تصویر P



شکل (۳). کدبرگردان MPEG-4 برای تصویر P

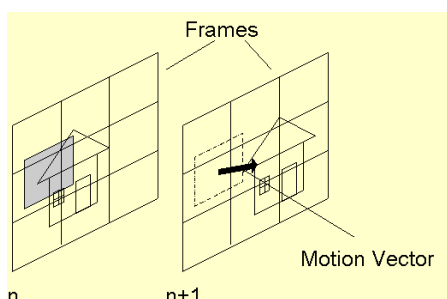
1- Additive-error
2- Data Partitioning



شکل (۶). مثالی از چگونگی محاسبه بردار حرکت و نشان دادن

نمونه‌ای از بلوک پنجره تخمین و بلوک جستجو [۱۳]

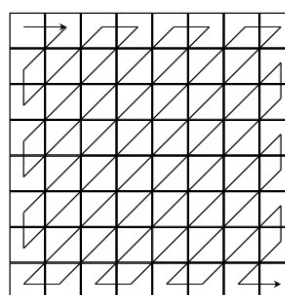
به‌عنوان مثال آنچه در توضیح آورده شد را می‌توان به‌صورت عملی در شکل (۷) مشاهده نمود.



شکل (۷). مثالی از محاسبه بردار حرکت بین دو فریم متوالی [۱۴]

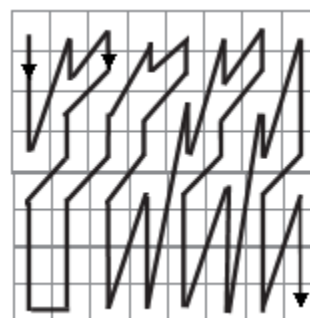
در مرحله بعد پس از تخمین کامل فریم مورد نظر و ساخته شدن ماتریس تخمین حرکت، تفاضل مقدار تخمین‌زده‌شده از فریم قبلی محاسبه می‌گردد. در انتها مقدار به‌دست‌آمده به بلوک DCT اعمال می‌گردد. پس از عبور از بلوک DCT مقدار به‌دست‌آمده با ۶۴ سطح کوانتیزه می‌گردد و در انتها بلوک‌های DCT را جهت ارسال به کانال با استفاده از روش کدکردن با طول متغیر^۲ کد می‌نماید [۱۵]. در گیرنده نیز همان‌طور که در شکل (۳) نشان داده شده است ابتدا عمل دی‌کوانتیزاسیون صورت می‌پذیرد و پس از آن عکس تبدیل DCT از اطلاعات دی‌کوانتیزه شده گرفته می‌شود و در مرحله آخر با استفاده از ماتریس تخمین حرکت فریم دریافتی را با فریم قبلی جمع کرده و به این ترتیب می‌توان فریم ارسالی را بازیابی کرد. در این مقاله با استفاده از روشی نوین برای تخمین و ارسال تابع چگالی احتمال بر مبنای کد هافمن و استفاده از شبکه عصبی به‌منظور حافظه کردن رشته اطلاعات ارسالی (در بخش ۶، توضیح داده شده است) برای افزایش میزان فشردگی بیت‌های اطلاعات پس از گذشتن از بلوک کدکننده با طول متغیر ارائه گردیده است و در مرحله بعد کدکننده کانال

این بدان معناست که پنجره‌ای با اندازه 16×16 برای جستجو انتخاب می‌گردد و طول پنجره تخمین 8×8 می‌باشد. به عبارت دیگر، در بازه پنجره جستجو دو فریم متوالی را با همدیگر مقایسه می‌نماید و کمترین مقدار به‌عنوان یکی از اعضا ماتریس تخمین حرکت ذخیره می‌گردد برای توضیح بیشتر به ارایه یک مثال از این نوع تخمین می‌پردازیم. همان‌طور که در (شکل ۶) نشان داده شده است، قرار است میزان حرکت یک شی در دو فریم که پشت سرهم قرار دارند با استفاده از محاسبه بردار حرکت تخمین‌زده شود. در شکل (۶) منظور از پنجره جستجو همان متغیر W می‌باشد که به اندازه 16×16 در نظر گرفته شده است. همچنین منظور از بلوک تطبیق^۱، همان پنجره تخمین است که به اندازه 8×8 در نظر گرفته شده است. همان‌طور که نشان داده شده است بلوک تخمین در محدوده‌ی پنجره جستجو، حرکت کرده و پس از حرکت، در تمام پنجره جستجو (که با پیاده‌سازی معادله (۱) به‌دست می‌آید) کمترین خطا بین دو بلوک، به‌عنوان محل جدید بلوک در فریم بعدی در نظر گرفته می‌شود (با استفاده از معادله (۲) محاسبه می‌شود). بنابراین با استفاده از این عملیات می‌توان به کمترین واریانس بین فریم تخمین‌زده‌شده و فریم اصلی دست یافت.



شکل (۴). نحوه انتخاب ضرایب DCT برای چیدن کنار هم در

روش زیگزاگ [۱۰].



شکل (۵). نحوه انتخاب ضرایب DCT برای چیدن کنار

هم در روش زیگزاگ متناوب [۱۱].

مرحله تعداد صفرها را می‌شمارد و این کار تا زمانی ادامه پیدا می‌کند که به درایه غیر صفر برسد و با این‌که تمام درایه‌های ماتریس DCT تمام شوند. پس از مشخص کردن تعداد صفرها به‌ازای تعداد تکرار صفر پس از درایه غیر صفر یک کد در جدول استاندارد تعیین شده برای MPEG آورده شده است. برای مثال برای ارسال عدد یک که پس از آن ۵ صفر وجود دارد به‌جای این‌که برای هر درایه ۸ بیت اختصاص داده شود که منجر به ارسال ۴۸ بیت می‌شود، تنها ۷ بیت که عبارتست از 000111s ارسال می‌گردد که متغیر s در بخش (۱-۱-۲) توضیح داده شده است.

این نوع کدگذاری بر اساس دو پارامتر در جدول استاندارد MPEG نشان داده می‌شود که به‌صورت میزان/ادامه ۳ نشان داده می‌شود [۵]. مقدار میزان برابر است با درایه غیر صفر ماتریس DCT و ادامه برابر است با تعداد صفر موجود پس از عدد انتخاب شده، قبل از این‌که به عدد غیر صفر دیگری برسیم، که این مقدار می‌تواند صفر باشد. به این معنی که عدد بعدی نیز مخالف صفر می‌باشد [۱۶].

1	-7	-8	0	2	0	0	0
3	4	-1	-3	-3	-1	0	0
-5	1	4	3	2	-1	-1	0
-2	-1	1	0	1	2	1	0
-1	0	-1	-1	-1	-1	0	0
1	1	0	0	0	0	0	0
3	-1	1	0	-1	0	0	0
0	-2	1	1	-1	0	0	0

شکل (۸). ضرایب ماتریس DCT کوانتیزه شده

۳- نحوه کدگذاری کانال

در این مقاله از کدکننده Reed-Solomon برای کدگذاری کانال استفاده شده است که این کد کلاسی از کدهای BCH دوره‌ای [۱۷-۱۸] خطی و غیرباینری که توسط میدان GF(q) ساخته می‌شوند، می‌باشد [۱۹]. دلیل استفاده از این نوع کدگذاری غیرباینری بودن آن می‌باشد. بدین‌صورت که اطلاعات هر بلوک را می‌توان بدون این‌که بلوک مورد نظر را به باینری تبدیل کنیم، مستقیماً کد نماییم و به این ترتیب سرعت کدگذاری فریم‌های ویدیویی بالاتر می‌رود. در این روش با فرض این‌که پیغام کدشده c(X) ارسال گردیده و پیام نویزی شده r(X) در گیرنده دریافت شود، داریم:

$$r(X) = r_0 + r_1X + \dots + r_{n-1}X^{n-1} \quad (3)$$

ثانویه‌ای با نرخ وابسته به تعداد بیت‌هایی که از تفاضل بین بیت‌های ساخته توسط بلوک کدکننده با طول متغیر و بیت‌های ساخته شده با استفاده از روش فشرده‌سازی معرفی شده در این مقاله به‌دست می‌آیند، مجدداً رشته بیت‌های ارسالی را کد می‌کند. در این مقاله سعی شده است رشته بیت‌ها را حافظه‌دار نمود تا بتوان چندین بار الگوریتم هافمن را با تابع چگالی‌های مختلف پیاده‌سازی کرد. در روش پیشنهادی به‌دلیل حافظه‌دار کردن این نوع کدگذاری به‌راحتی می‌توان رشته بیت‌ها را در گیرنده بازیابی کرد.

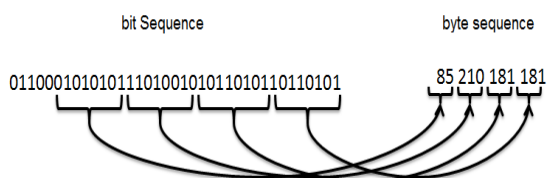
۲-۱-۵- کد کردن با طول متغیر^۱

کد کردن با طول متغیر آخرین مرحله قبل از ارسال اطلاعات مربوط به هر فریم می‌باشد. اساس این کدگذاری بدین صورت است که سعی در کاهش اطلاعات ارسالی دارد. این کدگذاری به دو صورت این عمل را انجام می‌دهد.

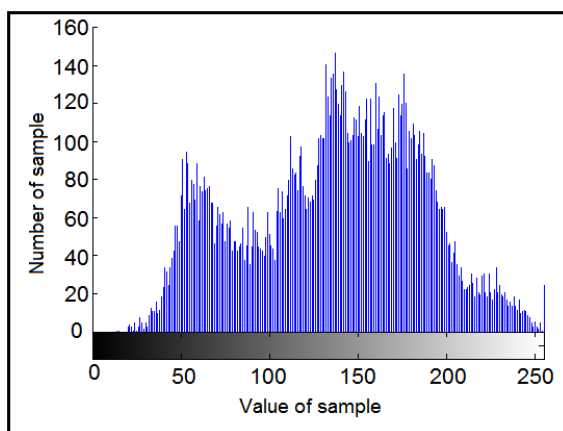
۲-۱-۵-۱- یکی از روش‌هایی که برای کاهش اطلاعات ارسالی در این نوع کدگذاری استفاده می‌گردد با توجه به زیاد بودن مقادیر یک در ماتریس DCT پس از کوانتیزاسیون زیاد می‌باشد که این امر در شکل ۸ به‌طور نمونه نشان داده شده است. بنابراین تعداد بیت کمی به آنها اختصاص می‌دهد. برای مثال تنها ۳ بیت برای ارسال عدد یک اختصاص می‌دهد که عبارتست از 11s که این کد با استفاده از جدول استاندارد تعیین شده برای MPEG آورده شده است [۱۵] در مثال قبل مقدار s با توجه به علامت درایه مورد نظر تعیین می‌گردد بدین صورت که اگر مثبت باشد s=0 و اگر منفی باشد s=1 در نظر گرفته می‌شود. به همین ترتیب تعداد بیت ارسالی برای اعداد بزرگتر از ۱ با توجه به این‌که نسبت به عدد ۱ کمتر ارسال می‌گردند بیشتر می‌باشند. برای مثال برای ارسال عدد ۷، ۱۱ بیت استفاده می‌گردد.

۲-۱-۵-۲- یکی دیگر از روش‌هایی که برای کاهش اطلاعات ارسالی در این نوع کدگذاری استفاده می‌گردد با در نظر گرفتن این واقعیت است که تعداد عناصر صفر نیز همانطور که در شکل ۸ نشان داده شده است در ماتریس DCT^۲ (تبدیل کسینوسی گسسته) پس از کوانتیزاسیون زیاد است. این نوع کدگذاری پس از چیده شده درایه‌های ماتریس DCT به‌صورت زیگزاگ جستجو می‌کند که آیا درایه بعدی صفر است یا خیر. در صورت صفر بودن به عدد بعد مراجعه می‌کند و در هر

نیاز به تعداد زیادی بیت داریم. به این منظور، ابتدا طول مورد نظری از بیت‌ها را از رشته اطلاعات جدا می‌کنیم یا این که منتظر می‌مانیم تا بافر به اندازه طول مورد نظر پر شود. پس از آن هر ۸ بیت از رشته مورد نظر را داخل یک بافر دیگر مطابق شکل (۹) می‌ریزیم. با این کار بافر به جای اعداد یک و صفر اعدادی بین صفر تا ۲۵۵ را در مبنای ده شامل می‌شود. در مرحله بعد تعداد تکرار این ۲۵۶ عدد در رشته مورد نظر محاسبه می‌گردد. بنابراین می‌توان هیستوگرام را بر حسب تعداد تکرار این ۲۵۶ عدد ساخته شده در رشته مورد نظر به دست آورد. به عنوان مثال شکل (۱۰) نمونه‌ای از هیستوگرام یک رشته بیت پس از تبدیل به رشته‌ای از بایت را نشان می‌دهد.



شکل (۹). نحوه جدا کردن بیت‌ها از رشته اطلاعات و ساختن رشته‌ای از بایت‌ها



شکل (۱۰). هیستوگرام تعداد تکرار نمونه‌ها در یک رشته اطلاعات

پس از این مرحله باید هیستوگرام فراوانی نمونه‌های ۸ بیتی را از هیستوگرام (شکل ۱۰) ساخته شده به دست آورد. برای به دست آوردن هیستوگرام فراوانی نمونه‌های ۸ بیتی از روی هیستوگرام باید شروط ۸، ۹ و ۱۰ ارضا شوند.

$$f_X(X) \geq 0 \quad (۸)$$

$$f_X(X) \leq 1 \quad (۹)$$

$$\sum_{-\infty}^{+\infty} f_X(X) = 1 \quad (۱۰)$$

در شروط ۸، ۹ و ۱۰، مقدار تابع هیستوگرام ساخته شده است که نشان‌دهنده تعداد تکرار اعداد بین صفر تا ۲۵۵

این چندجمله‌ای خطا $e(X)$ و رابطه آن با پیام ارسالی و دریافتی در معادله (۴) آورده شده است.

$$e(X) = e_0 + e_1X + \dots + e_{n-1}X^{n-1} \quad (۴)$$

$$r(X) = c(X) + e(X)$$

بنابراین با توجه به معادله (۴) می‌توان نشان داد که $c(\alpha^i) + e(\alpha^i) = e(\alpha^i)$

در این نوع کدگذاری برای پیدا کردن خطا علاوه بر محل خطا باید مقدار خطا نیز محاسبه گردد. با توجه به مقدار متغیر t و همچنین رابطه $\beta_i = \alpha^i$ می‌توان با استفاده از معادله (۵) محل و مقدار خطا را به دست آورد. این نوع کدگذاری قادر است هر الگوی خطا با طول t یا کمتر را تصحیح نماید.

$$1 = r(\alpha) = e(\alpha) = e_{j_1}\beta_1 + e_{j_2}\beta_2 + \dots + e_{j_t}\beta_t$$

$$1 = r(\alpha^2) = e(\alpha^2) = e_{j_1}\beta_1^2 + e_{j_2}\beta_2^2 + \dots + e_{j_t}\beta_t^2 \quad (۵)$$

$$2t = r(\alpha^{2t}) = e(\alpha^{2t}) = e_{j_1}\beta_1^{2t} + e_{j_2}\beta_2^{2t} + \dots + e_{j_t}\beta_t^{2t}$$

در ساده‌ترین حالت، در این کدگذاری با فرض $t=1$ مقدار و محل خطا با استفاده از معادله (۶) محاسبه می‌گردد.

$$\alpha^{j_1} = \frac{S_2}{S_1} \Rightarrow \text{Error Place} \quad (۶)$$

$$e_{j_1} = \frac{S_1^2}{S_2} \Rightarrow \text{Error Value}$$

رابطه نرخ کد کننده کانال که در معادله (۷) نشان داده شده است که واحد آن بیت بر ثانیه می‌باشد.

$$R_c = \frac{k}{n} \quad (۷)$$

در رابطه k ، n تعداد بیت‌های مربوط به اطلاعات فریم که قرار است کد شوند می‌باشد. n طول رشته ساخته شده برای ارسال است که شامل پیام اصلی و اطلاعات اضافه شده توسط انکدر کانال می‌باشد. انکدر با مولفه‌های فوق قادر است به میزان $\frac{n-k}{2}$ خطا را تشخیص و تصحیح کند، بنابراین هرچه میزان n (تعداد بیت‌های توازن) بیشتر باشد تعداد بیشتری از بیت‌های پیام را می‌توان تصحیح نمود.

۴- نحوه به دست آوردن هیستوگرام فراوانی

نمونه‌های ۸ بیتی برای یک رشته بیت با طول

دلخواه

برای به دست آوردن هیستوگرام فراوانی نمونه‌های ۸ بیتی

۵- شبکه عصبی مصنوعی

شبکه عصبی مصنوعی از یکسری لایه‌ها و گره‌ها تشکیل شده است. هر شبکه الزاما دارای لایه ورودی و خروجی می‌باشد [۲۱-۲۰] در هر شبکه می‌توان به دلخواه تعدادی لایه فرعی اضافه نمود. هر لایه شامل تعداد دلخواهی گره می‌باشد که تحت عنوان نرون‌ها شناخته می‌شوند. ارتباط بین هر لایه توسط لینک‌هایی که به نرون‌ها متصل می‌شوند برقرار می‌گردد. به این لینک‌ها اصطلاحا وزن‌های شبکه می‌گویند.

در شکل (۱۲) ساختاری از یک شبکه عصبی با هفت ورودی و یک لایه فرعی و سه خروجی نشان داده شده است. لازم به ذکر است که علاوه بر وزن‌ها، هر نرون متاثر از کمیت مستقل دیگری با نام ارباب نیز می‌باشد. نحوه تاثیرپذیری مقادیر وزن‌ها و ارباب‌ها در نرون‌های هر لایه، به نرون‌های لایه بعدی توسط تابعی تحت عنوان تابع انتقال، مشخص می‌شود. تابع انتقال در هر لایه می‌تواند خطی و یا غیرخطی باشد که این امر توسط کاربری که قرار است با شبکه عصبی کار کند؛ تعیین می‌گردد. بر اساس مطالبی که عنوان گردید، وزن‌ها و ارباب‌های مربوط به هر نرون، عامل مهم و حائز اهمیت در شبکه می‌باشند که مقادیر آنها در نتایج خروجی از شبکه عصبی تاثیر گذار می‌باشد. لذا در روش‌های متعددی که جهت بهبود دقت شبکه‌های عصبی به کار رفته است، سعی بر انتخاب مقادیر مناسب این وزن‌ها و ارباب‌ها به گونه‌ای که کمترین خطای شبکه را در بر داشته باشد، شده است. به این عمل اصطلاحا آموزش شبکه گفته می‌شود.

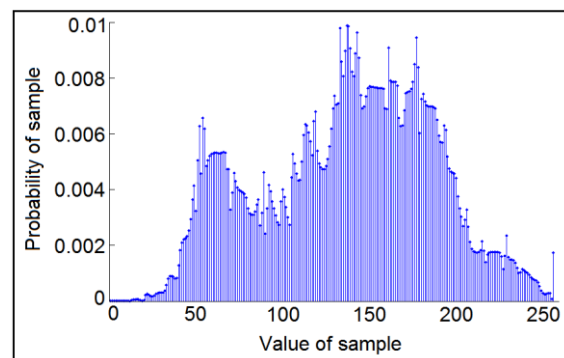
۵-۱- نحوه استفاده از شبکه عصبی برای توصیف منحنی هیستوگرام فراوانی نمونه‌های ۸ بیتی

در این قسمت می‌خواهیم تابعی بر منحنی چگالی احتمال (شکل ۱۱) با استفاده از شبکه عصبی منطبق کنیم [۲۲]. برای این کار ابتدا یک شبکه عصبی با یک لایه پنهان که آموزش ندیده است، ایجاد می‌شود. سپس تمام وزن‌ها و با یاس‌های شبکه به عنوان متغیرهای بهینه‌سازی معرفی می‌شوند. ورودی شبکه عصبی مقادیر محور افقی منحنی هیستوگرام فراوانی نمونه‌های ۸ بیتی (اعداد صحیح بین صفر تا ۲۵۵) و خروجی شبکه عصبی مقادیر محور عمودی تابع چگالی احتمال (احتمال رخداد نمونه‌ها) معرفی می‌گردند. تابع هدف در شبکه عصبی به کار گرفته شده بر اساس میانگین مربعات خطا بین احتمال رخداد نمونه‌های تابع چگالی احتمال و مقادیر حساب شده با توجه به وزن‌ها و ارباب‌های شبکه عصبی معرفی می‌گردند.

می‌باشد. با بررسی هیستوگرام، می‌توان نشان داد که شرط (۸) ارضا می‌شود، زیرا تعداد تکرار یک عدد همواره عدد مثبتی خواهد بود و در صورت نبودن یک عدد مقدار آن در نمودار هیستوگرام (شکل ۱۰) برابر با صفر خواهد شد. اما تابع هیستوگرام به تنهایی شروط ۲ و ۳ را ارضا نمی‌کند. برای ارضا کردن شروط ۲ و ۳ باید تعداد تکرار برای هر نقطه را بر مجموع کل تکرارهای موجود تقسیم کرد (معادله ۱۱). در این صورت شروط ۲ و ۳ نیز ارضا می‌گردند. با این کار توانستیم منحنی هیستوگرام فراوانی نمونه‌های ۸ بیتی (شکل ۱۱) را با استفاده هیستوگرام (شکل ۱۰) به دست آوریم.

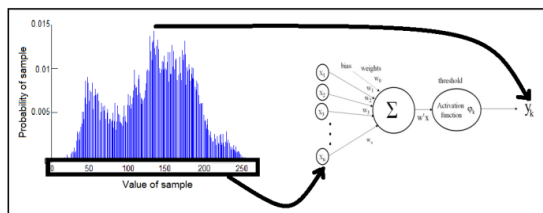
$$f_X(X_i) = \frac{f_X(X_i)}{\sum_{k=0}^{255} f_X(X_k)} \quad (11)$$

تا این مرحله توانستیم هیستوگرام فراوانی نمونه‌های ۸ بیتی مربوط به تعداد تکرار اعداد استفاده شده در یک رشته را به دست آوریم. اما با توجه به این که هیستوگرام فراوانی نمونه‌های ۸ بیتی برای هر رشته شکلی متفاوت از رشته دیگر دارد، برای کد کردن این رشته‌ها با الگوریتم هافمن باید تابع چگالی احتمال نیز همراه با رشته اطلاعات مربوط به آن هیستوگرام فراوانی نمونه‌های ۸ بیتی ارسال گردد. یکی از روش‌هایی که می‌توان برای ارسال هیستوگرام فراوانی نمونه‌های ۸ بیتی استفاده نمود، این است که بر روی آن نموداری منطبق کرده و اطلاعات آن نمودار را به همراه رشته اطلاعات ارسال نمود. اکنون این سوال پیش می‌آید که با توجه به تغییرات سریع و غیر خطی هیستوگرام فراوانی نمونه‌های ۸ بیتی (شکل ۱۱) چه نموداری می‌توان بر روی آن منطبق کرد به وجهی که کمترین خطا را با هیستوگرام فراوانی نمونه‌های ۸ بیتی مورد نظر داشته باشد. برای این کار می‌توان از شبکه عصبی مصنوعی کمک گرفت که در بخش ۵، به صورت خلاصه به توضیح آن می‌پردازیم.

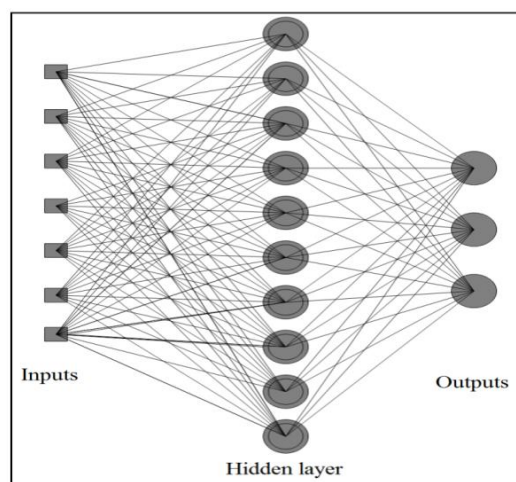


شکل (۱۱). منحنی هیستوگرام فراوانی نمونه‌های ۸ بیتی

هدف رسید تمامی مقادیر احتمال رخداد نمونه‌ها به عنوان نقاط آموزشی به شبکه عصبی وارد می‌گردند و در نتیجه پس از چندین بار به‌روز شدن مقادیر وزن‌ها و ارباب‌ها، مقادیر تابع به-دست‌آمده با استفاده از شبکه عصبی بسیار به منحنی چگالی احتمال نزدیک می‌شود.



شکل (۱۳). نمایش ورودی و خروجی به‌کارگرفته‌شده در شبکه عصبی



شکل (۱۲). نمایش ساختار یک شبکه عصبی چندلایه

۵-۲- پیاده‌سازی شبکه عصبی برای هیستوگرام

فراوانی نمونه‌های ۸ بیتی

به دلیل متغیر با زمان بودن منحنی‌های چگالی احتمال و همچنین به دلیل افت و خیزهای بالا در این منحنی‌ها نمی‌توان توابعی را با استفاده از روش‌های کلاسیک برای منطبق کردن بر روی این منحنی‌ها به‌دست آورد. بنابراین در این مقاله از شبکه عصبی برای منطبق ساختن تابعی بر روی منحنی چگالی احتمال استفاده نموده‌ایم. اگر بتوانیم تابعی بر منحنی چگالی احتمال منطبق سازیم، قادر به ذخیره‌سازی مقادیر تابع محاسبه‌شده، خواهیم بود که در این مقاله از این روش برای حافظه‌دار کردن توابع چگالی احتمال استفاده شده است. برای مثال اگر شبکه عصبی ساخته‌شده با ۶۰ نرون در لایه پنهان که دارای ۱۲۰ وزن و ۶۱ ارباب باشد^۱ را برای رشته اطلاعاتی که دارای منحنی چگالی احتمالی شبیه به شکل (۱۱) است آموزش دهیم، خروجی در بهترین حالت با فرض (خطای میانگین مربعی کمتر از 10^{-10}) در شکل‌های (۱۴ و ۱۵) نشان داده شده است.

در شکل (۱۴) تابع هدف و مقادیر خروجی به‌دست‌آمده با استفاده از شبکه عصبی با یکدیگر رسم شده‌اند. همان‌طور که در این شکل (۱۴) دیده می‌شود مقادیر خروجی با خطای بسیار کمی بر روی مقادیر تابع هدف منطبق گردیده‌اند. شکل (۱۵) مقدار رگرسیون^۲ را نشان می‌دهد که دلیلی بر صحت میزان خطای ناچیز داده‌های خروجی می‌باشد. البته باید به این نکته توجه کرد که در مواقعی که نمودار با شیب تند نزول می‌کند و مقادیر به سمت صفر میل می‌کنند، ممکن است که تابع تقریب-

بنابراین در هر مرحله از بهینه‌سازی، مقادیر وزن‌ها و ارباب‌های شبکه به‌گونه‌ای به‌روز می‌شوند که خطای شبکه کاهش یابد. در این مقاله برای منطبق کردن تابعی بر نمودار چگالی احتمال از شبکه عصبی پرسپترون با یک لایه پنهان که دارای ۶۰ نرون است استفاده شده است. تابع انتقال نرون‌های لایه پنهان tansig است و تابع انتقال لایه خروجی purelin می‌باشد. همچنین لایه ورودی و خروجی تک نرونی هستند و تمامی نرون‌های هر سه لایه دارای ارباب می‌باشند. شکل (۱۳) به‌منظور نشان‌دادن ورودی و خروجی استفاده‌شده در شبکه عصبی، نمایش داده شده است. ورودی شبکه عصبی، مقادیر محور افقی و تابع هدف شبکه عصبی، مقادیر عمودی متناظر با مقادیر افقی می‌باشد. وزن‌های شبکه تا زمانی که خطا از 10^{-10} کمتر شود، به‌روز می‌شوند. باید به این نکته اشاره نمود که نوع تابع انتقال با روش Leave-One-Out [۲۴-۲۳]، پس از امتحان بر روی چندین تابع چگالی احتمال انتخاب شده‌اند. در این روش قصد ما در آموزش دادن شبکه عصبی بر خلاف آموزش‌های معمول می‌باشد. به این دلیل که در آموزش‌های معمول درصدی از اطلاعات به‌عنوان داده‌های آزمون کنار گذاشته می‌شوند و در انتهای هر بار آموزش و به‌روز شدن وزن‌ها و ارباب‌ها، داده‌های آزمون به‌عنوان ورودی به شبکه عصبی وارد می‌گردند و خروجی‌های به‌دست‌آمده با مقادیر صحیح مقایسه می‌شوند. اگر خطا از حد مشخص‌شده کمتر باشد، وزن‌ها و ارباب‌های به‌روز-شده به‌عنوان وزن‌ها و ارباب‌های شبکه ثبت می‌گردند. ولی اگر خطا بیشتر از حد مشخص شده باشد آموزش دوباره تکرار می‌شود. این کار از آموزش بیش از حد جلوگیری می‌کند.

اما در این مقاله بر خلاف روش‌های معمول سعی بر این است که شبکه کاملاً بر مقادیر احتمال رخداد نمونه‌ها منطبق گردد و آموزش بیش از حد رخ دهد. برای این‌که بتوان به این

۱- تعداد ۶۰ بایاس برای لایه پنهان و یک بایاس برای لایه خروجی

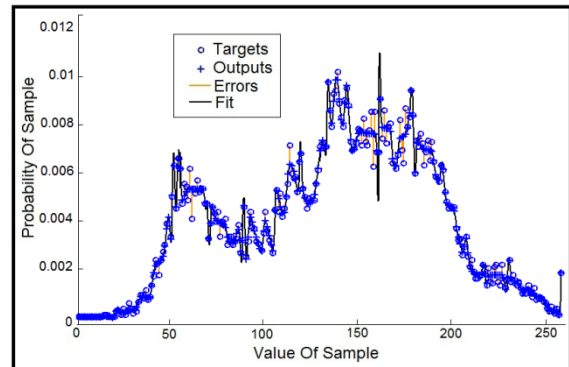
۲- میزان پراکندگی مقادیر هدف از مقادیر خروجی

در بخش‌های ۲-۵ و ۱-۵، می‌توان گفت که هیستوگرام فراوانی نمونه‌های ۸ بیتی را با استفاده از شبکه عصبی مصنوعی و در مواردی که بیشتر مقادیر هیستوگرام فراوانی نمونه‌های ۸ بیتی در چند نمونه محدود شده است؛ با استفاده از این چند نمونه (در بخش ۷-۱، توضیح داده می‌شود) می‌توان توصیف کرد. بنابراین، از این به بعد می‌توان هیستوگرام فراوانی نمونه‌های ۸ بیتی را با وزن‌ها و ارب‌های به‌دست‌آمده از آموزش شبکه عصبی و یا تعداد محدودی عدد توصیف نمود. همچنین رفتار هیستوگرام فراوانی نمونه‌های ۸ بیتی را با استفاده از مجموعه وزن‌ها و ارب‌های شبکه عصبی و یا تعداد محدودی عدد می‌توان ذخیره‌سازی کرد. برای ذخیره هر کدام از وزن‌ها و ارب‌ها و یا اعداد توضیح داده شده ۲ بایت بیشتر نیاز نمی‌باشد. در ادامه مزیت حافظه‌دار کردن رشته اطلاعات را خواهیم دید.

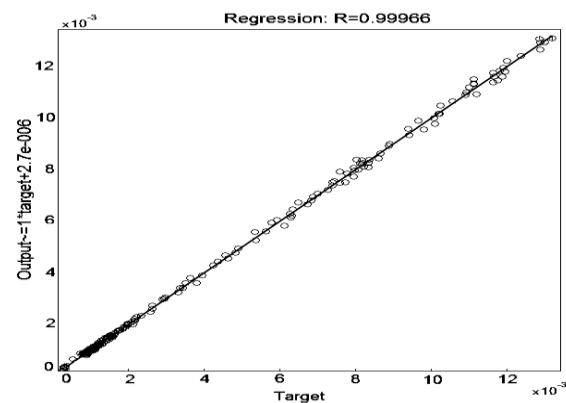
۶-۱- مزیت حافظه‌دار کردن رشته اطلاعات

برای حافظه‌دار کردن رشته اطلاعات می‌توان الگوریتم هافمن را چندین بار بر روی رشته اطلاعات پیاده‌سازی کرد. بدین صورت که در ابتدا هیستوگرام فراوانی نمونه‌های ۸ بیتی رشته اطلاعاتی مورد نظر تخمین زده می‌شود. این عملیات با استفاده از شبکه عصبی صورت می‌گیرد (منحنی‌ای را با استفاده از هیستوگرام فراوانی نمونه‌های ۸ بیتی تخمین می‌زند به طوری که تا جای ممکن کمترین خطا بین نمونه‌های تخمین‌زده‌شده و نمونه‌های هیستوگرام فراوانی نمونه‌های ۸ بیتی وجود داشته باشد). در مرحله بعد رشته مورد نظر با استفاده از هیستوگرام فراوانی نمونه‌های ۸ بیتی تخمین زده شده، با الگوریتم هافمن فشرده می‌گردد. در مرحله بعد، مجموعه وزن‌های شبکه عصبی به ابتدای رشته فشرده‌شده اضافه می‌گردد. تا اینجا تنها یک مرتبه عملیات فشرده‌سازی صورت گرفته و رشته مورد نظر، هنوز بدون حافظه می‌باشد. در مرحله بعد، رشته فشرده‌شده به همراه وزن‌های شبکه عصبی که با استفاده از آن‌ها منحنی چگالی احتمال تخمین زده شده است به‌عنوان رشته جدید در نظر گرفته‌شده و هیستوگرام فراوانی نمونه‌های ۸ بیتی جدیدی با استفاده از مقادیر جدید به‌دست می‌آید و مشابه مرحله قبل ابتدا تابع تخمین‌زده‌شده و مجدداً با استفاده از الگوریتم هافمن فشرده می‌شود و وزن‌های شبکه عصبی جدید به انتهای رشته ثانویه ارسال می‌گردد. با تکرار این روش می‌توان حجم رشته اطلاعات ارسالی را بسیار کم نمود. در بخش (۷-۱)، به توضیح بیشتر تکنیکی این عملیات پرداخته می‌شود.

زده مقادیری کمتر از صفر را برای این نقاط تقریب بزنند که شرط ۸ را نقض می‌کند. در این مواقع مقادیر منفی با کمترین مقدار مثبت تابع تخمین‌زده‌شده جایگزین می‌گردند. زیرا نقاطی که با مقادیر منفی تخمین زده شده‌اند دارای مقادیر ناچیزی هستند و لذا می‌توان آنها را با کمترین مقدار مثبت تابع تخمین‌زده‌شده جایگزین کرد.



شکل (۱۴). رسم همزمان مقادیر تابع هدف همراه با مقادیر خروجی به‌دست‌آمده با شبکه عصبی



شکل (۱۵). رسم میزان پراکندگی مقادیر هدف از مقادیر خروجی

۶- مفهوم حافظه‌دار کردن رشته اطلاعات

در اولین گام باید به معنی حافظه‌دار کردن رشته اطلاعات بپردازیم. دلیل اطلاق چنین نامی به این خاطر است که رشته اطلاعات چندین مرحله فشرده خواهد شد و در عملیات وافشرده‌سازی^۱ در هر مرحله، برای وافشرده کردن به اطلاعاتی که از رشته وافشرده‌شده در مرحله قبل به‌دست‌آمده، نیاز است. منظور از این اطلاعات همان وزن‌های شبکه عصبی می‌باشد. بنابراین در هر مرحله رشته مورد نظر بسته به نوع هیستوگرام فراوانی نمونه‌های ۸ بیتی که دارد با استفاده از ضرایب وزنی متفاوتی توصیف می‌شود. بنابر آنچه گفته شد می‌توان رشته ساخته‌شده را حافظه‌دار در نظر گرفت. با توجه به مطالب گفته‌شده

۷- الگوریتم پیشنهادی

۷-۱- کد کننده الگوریتم پیشنهادی

فلوچارت کد کننده الگوریتم پیشنهادی در شکل (۱۶) نشان داده شده است. اساس روش پیشنهادی به این صورت می باشد که پس از پیش پردازش هایی که روی اطلاعات خروجی از بلوک VLC صورت می پذیرد، اطلاعات با استفاده از کد هافمن کد می گردند و سپس در مرحله بعد کد کننده کانال ثانویه ای با نرخ وابسته به تعداد بیت هایی که از تفاضل بین بیت های ساخته توسط بلوک VLC و بیت های ساخته شده با استفاده از کد هافمن به دست می آیند، مجددا رشته بیت های ارسالی را کد می کند. مراحل انجام این الگوریتم به شرح زیر است.

- مرحله ۱: رشته بیت اطلاعات را در بسته های ۸ بیتی جدا نموده و سپس از مبنای دودویی به مبنای دهدهی تبدیل کنیم (بخش ۴)
- مرحله ۲: منحنی هیستوگرام را بر اساس تعداد تکرار اعداد به دست آمده از مرحله ۱ (رنج این اعداد بین صفر تا ۲۵۵ می باشد). به دست می آوریم. (بخش ۴)
- مرحله ۳: به دست آوردن منحنی چگالی احتمال با استفاده از منحنی هیستوگرام، با ارضا کردن شروط ۸، ۹ و ۱۰. (بخش ۴)
- مرحله ۴: اگر تعداد تکرار الگوریتم با استفاده از شبکه عصبی از ۴ مرتبه بیشتر باشد به مرحله ۱۶ پرش کن در غیر این صورت استفاده از شبکه عصبی برای منطبق کردن تابعی بر روی منحنی چگالی احتمال صورت می گیرد. (بخش ۵)
- مرحله ۵: اجرای الگوریتم هافمن با استفاده از هیستوگرام فراوانی نمونه های ۸ بیتی تخمین زده شده در مرحله ۴.
- مرحله ۶: تبدیل وزن ها و اربها از مبنای دهدهی به مبنای دودویی و اضافه کردن آنها به انتهای رشته اطلاعات فشرده شده جهت ارسال.
- مرحله بعد با توجه به توضیح زیر در رابطه با ویژگی کد هافمن ادامه داده می شود:

با توجه به این که کد هافمن در جهتی رشته اطلاعات را کد می کند که تعداد صفرهای ساخته شده ماکزیمم باشد [۲۶-۲۵]، اگر دوباره عملیات مرحله ۱ روی رشته بیت ساخته شده از

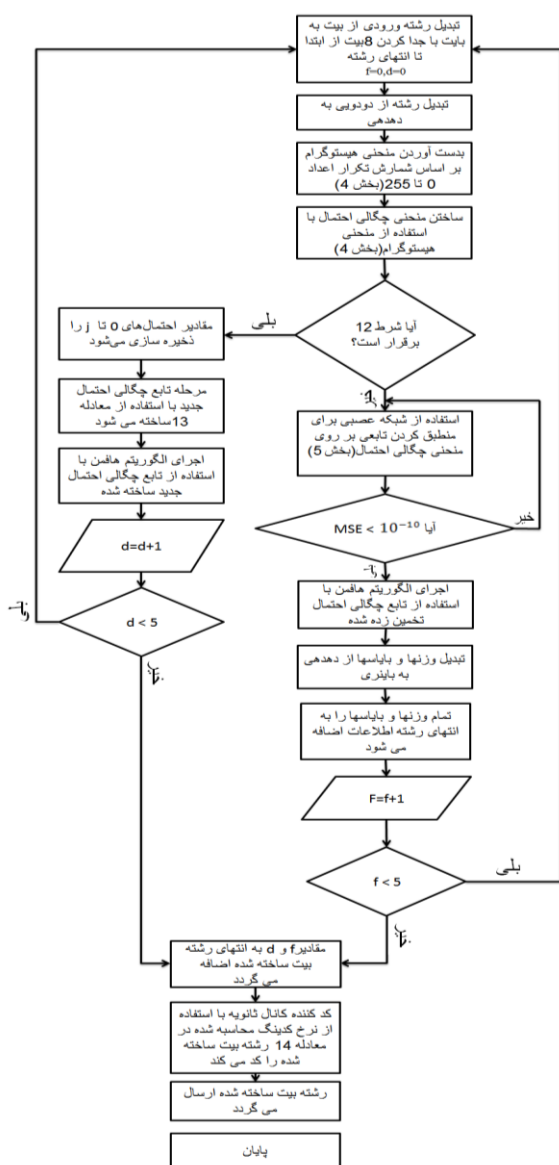
خروجی کد هافمن صورت پذیرد، همان طور که در شکل های (۱۷ و ۱۸) نشان داده شده است. شکل (۱۸)، بزرگنمایی شکل (۱۷) می باشد که برای بهتر نشان دادن مطلب رسم گردیده است. بیشتر اعداد در رنج های پایین تکرار شده اند و اعداد بزرگتر تنها دارای تکرارهای معدودی هستند (البته باید به این نکته اشاره نمود که این عملیات بر روی داده های فراوانی صورت پذیرفته و صحت گفته فوق تایید شده است). این موضوع به صورت تجربی به دست آمده است. بنابراین، شرطی برای برقراری این موضوع چک می گردد و در صورت برآورده شدن به مرحله بعد می رود. باید به این نکته اشاره نمود اگر شروط مورد نظر برقرار نباشد میزان فشرده سازی برابر با زمانی می شود که یکبار از الگوریتم هافمن استفاده شود. بنابراین، در صورت برقرار نشدن شروط مورد نظر، میزان فشرده سازی روش فوق به صورتی خواهد بود که روش مورد نظر موجب افزایش رشته بیت نسبت به زمانی که یکبار از الگوریتم هافمن استفاده شود، نمی گردد.

- مرحله ۷: رشته بیت ساخته شده در مرحله ۶ و یا ۱۴ را مجددا در بسته های ۸ بیتی جدانموده، سپس از مبنای دودویی به مبنای دهدهی تبدیل می کنیم (بخش ۴)
- مرحله ۸: منحنی هیستوگرام را بر اساس تعداد تکرار اعداد به دست آمده از مرحله ۷ (رنج این اعداد بین صفر تا ۲۵۵ می باشد). به دست می آوریم. (بخش ۴)
- مرحله ۹: به دست آوردن منحنی چگالی احتمال با استفاده از منحنی هیستوگرام، با ارضا کردن شروط ۸، ۹ و ۱۰. (بخش ۴)
- مرحله ۱۰: در صورت برقرار بودن شرط ۱۲ به مرحله بعد می رویم و در غیر این صورت به مرحله ۴ پرش می کنیم.

$$\sum_{k=0}^j f_X(X_k) > 0.9 ; j = 1, 2, 3, 4 \quad (12)$$

- مرحله ۱۱: مقادیر احتمال های 0 تا z را ذخیره سازی می شود.
- مرحله ۱۲: در این مرحله هیستوگرام فراوانی نمونه های ۸ بیتی جدیدی می سازیم. با توجه به این مطلب که بیشتر مقادیر هیستوگرام فراوانی نمونه های ۸ بیتی را که در چند نمونه اولیه قرار داشت ذخیره کرده ایم. بنابراین به دلیل بسیار ناچیز بودن دیگر مقادیر هیستوگرام فراوانی نمونه های ۸ بیتی فرض

تعداد بیت ساخته شده پس از فشرده سازی با استفاده از کد هافمن می باشد.



شکل (۱۶). فلوچارت کدکننده الگوریتم پیشنهادی

$$R_c' = \frac{N_b'}{N_b} \quad (14)$$

در معادله ۱۴ منظور از R_c' نرخ کدگذاری کانال ثانویه می باشد و منظور از N_b' تعداد بیت ساخته شده توسط بلوک VLC می باشد (تعداد بیتی که در نهایت ساخته می شود باید به همین تعداد باشد) و منظور از N_b تعداد بیت ساخته شده پس از فشرده سازی با استفاده از کد هافمن می باشد.

می شود تمام این نمونه ها دارای مقادیر هیستوگرام فراوانی نمونه های ۸ بیتی برابر هستند. بنابراین مقدار به دست آمده با استفاده از معادله (۱۳) به هر یک از آنها نسبت داده می شود.

$$f_X(X_i) = \frac{1 - \sum_{k=0}^j f_X(X_k)}{256 - j}; \quad (13)$$

$$i = j+1, j+2, \dots, 255$$

مرحله ۱۳: اجرای الگوریتم هافمن با استفاده از هیستوگرام فراوانی نمونه های ۸ بیتی ساخته شده در مرحله ۱۲.

مرحله ۱۴: مقدار z عدد اولیه هیستوگرام فراوانی نمونه های ۸ بیتی ذخیره شده در مرحله ۱۱ و همچنین مقدار z نیز از مبنای دهدهی به مبنای دودویی تبدیل شده و به انتهای رشته اطلاعات فشرده شده جهت ارسال اضافه می گردند.

مرحله ۱۵: اگر تعداد تکرار الگوریتم ساختن هیستوگرام فراوانی نمونه های ۸ بیتی بدون استفاده از شبکه عصبی از ۴ مرتبه بیشتر باشد به مرحله ۱۶ پرش کن در غیر این صورت دوباره به مرحله ۷ باز می گردیم.

مرحله ۱۶: مقدار f و d (به ازای هر مقدار ۳ بیت اختصاص داده می گردد. به دلیل این که بیشترین مقدار روش پیشنهادی ۵ بار تکرار است که با ۳ بیت می توان نمایش داد) و همچنین ترتیب اجرای آنها به انتهای رشته بیت ساخته شده، اضافه می گردد (به این معنا که آیا هیستوگرام فراوانی نمونه های ۸ بیتی در هر مرحله با استفاده از شبکه عصبی تخمین زده شده یا خیر). به این منظور به هر روش یک کد ۲ بیتی داده می شود و پس از مقدار f و d نحوه ترتیب آنها نیز آورده می شود، که این کد به صورت ۱۰ و ۰۱ می باشد.

مرحله ۱۷: در این مرحله کدکننده کانال ثانویه با استفاده از نرخ کدگذاری محاسبه شده در معادله (۱۴) رشته بیت ساخته شده را کد می کند و نرخ کدگذاری نیز به انتهای رشته اضافه می گردد. در معادله ۱۴ منظور از R_c' نرخ کدگذاری کانال ثانویه می باشد و منظور از N_b' تعداد بیت ساخته شده توسط بلوک VLC می باشد (تعداد بیتی که در نهایت ساخته می شود باید به همین تعداد باشد) و منظور از N_b

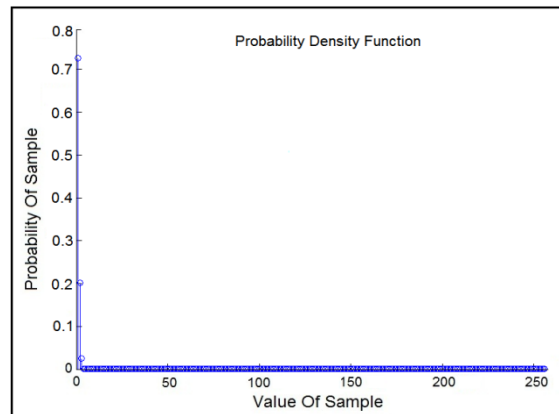
نپذیرفته باشد) به اندازه Q بیت به دست آمده با استفاده از معادله (۱۵) از انتهای رشته اطلاعات جدا می‌کنیم.

$$Q = (2 + 3 * N_n) * \frac{2 * \text{byte}}{\text{neuron}} * \frac{8 * \text{bit}}{\text{byte}} \quad (15)$$

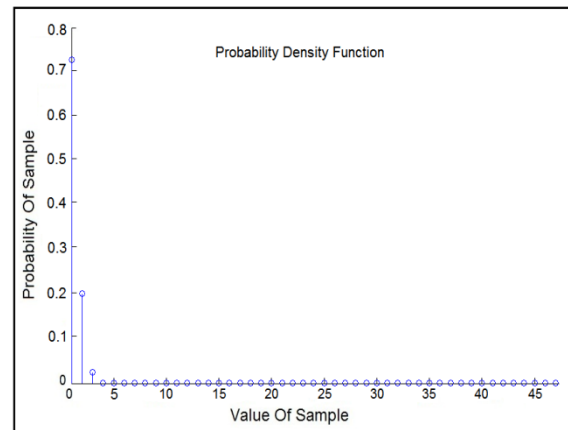
در معادله (۱۵)، منظور از N_n تعداد نرون لایه پنهان و به دلیل این که لایه خروجی تک‌نرونی است تعداد وزن‌های لایه پنهان برابر است با $2 * N_n$. همچنین هر لایه یک اریب نیز دارد بنابراین، به تعداد N_n اریب در لایه پنهان داری m ، برای ارسال وزن‌ها و اریب لایه پنهان مقدار $3 * N_n$ عدد باید ارسال گردد. برای لایه خروجی به دلیل این که تک‌نرونی است، فقط ۲ عدد که یکی برای وزن خروجی و دیگری برای اریب است ارسال می‌گردد. همچنین برای ارسال هر وزن و اریب ۲ بایت^۱ در نظر گرفته می‌شود.

- مرحله ۴: با استفاده از وزن‌ها و اریب‌های تولید شده از مرحله ۳ و با استفاده از شبکه عصبی هیستوگرام فراوانی نمونه‌های ۸ بیتی دوباره بازسازی می‌شود.
- مرحله ۵: با توجه به هیستوگرام فراوانی نمونه‌های ۸ بیتی به دست آمده رشته اطلاعات با استفاده از الگوریتم هافمن دیکد می‌گردد.
- مرحله ۶: مجدداً بررسی می‌کنیم که آیا در کدکننده هیستوگرام فراوانی نمونه‌های ۸ بیتی با استفاده از شبکه عصبی تخمین زده شده یا خیر. اگر با استفاده از شبکه عصبی تخمین زده شده بود که به مرحله ۳ پرش می‌کنیم با استفاده از دو کد ۰۱ و ۱۰ مشخص می‌گردد که در بخش ۷-۱، مرحله ۱۶ توضیح داده شده است، (برای تمام فریم‌های ارسالی تنها برای مرتبه اول، شبکه عصبی مورد استفاده قرار گرفته است.) در غیر این صورت به مرحله بعد می‌رویم.
- مرحله ۷: در این مرحله به تعداد P بیت که از معادله (۱۶) محاسبه می‌گردد، از انتهای رشته بیت ساخته شده از مرحله قبل جدا می‌کنیم. در معادله ۱۶ منظور از Z تعداد اعدادی که در معادله (۱۲) صدق می‌کند را نشان می‌دهد.

- مرحله ۱۸: رشته بیت ساخته شده ارسال می‌گردد.



شکل (۱۷). منحنی تابع چگالی احتمال جدید به دست آمده از منحنی شکل (۱۱) پس از پیاده‌سازی کد هافمن بر روی رشته اطلاعات

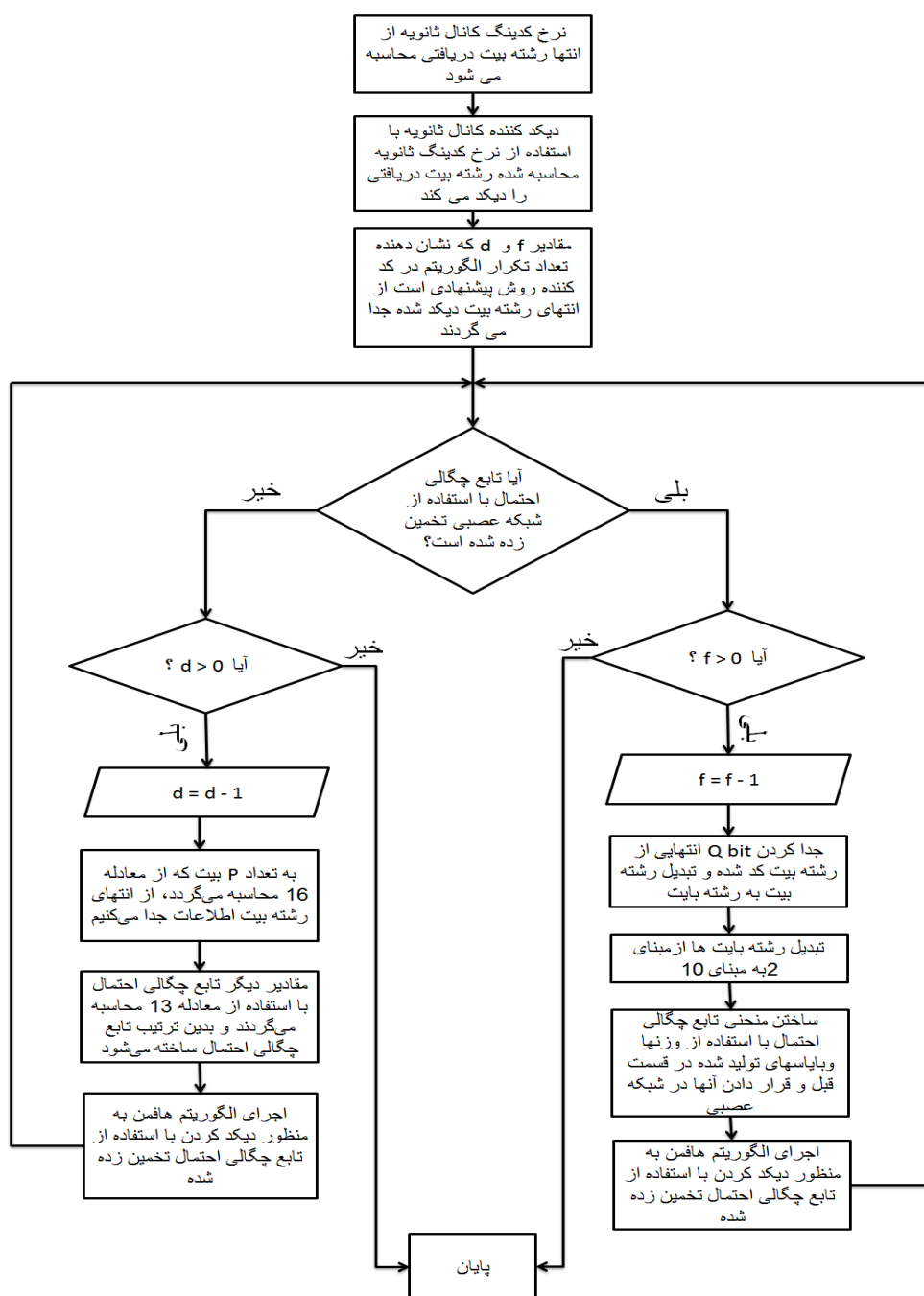


شکل (۱۸). درشت‌نمایی شکل (۱۷) در محدوده بین نمونه‌های صفر تا ۴۷

۷-۲- کدبرگردان الگوریتم پیشنهادی

فلوچارت کدبرگردان الگوریتم پیشنهادی در شکل ۱۹ نشان داده شده است. مراحل دیکد کردن اطلاعات به شرح زیر می‌باشد.

- مرحله ۱: ابتدا نرخ کدگذاری کانال ثانویه از انتهای رشته دریافتی محاسبه می‌گردد و سپس دیکد کننده کانال با استفاده از این نرخ کدگذاری اطلاعات را دیکد می‌نماید.
- مرحله ۲: مقادیر f و d از انتهای رشته بیت دیکد شده به همراه ترتیب اجرای آنها محاسبه می‌گردند.
- مرحله ۳: در صورتی که تابع چگالی احتمال با استفاده از شبکه عصبی تخمین زده شده باشد (شرط ۱۲ صورت



شکل (۱۹). فلوچارت کدبرگردان الگوریتم پیشنهادی

- ترتیب تابع چگالی احتمال ساخته می شود.
- مرحله ۹: با توجه به تابع چگالی احتمال به دست آمده رشته اطلاعات با استفاده از الگوریتم هافمن دیکد می شود.
 - مرحله ۱۰: این الگوریتم به تعداد $f+d$ مرتبه تکرار می گردد. اگر تعداد تکرار کمتر از $f+d$ باشد به مرحله ۶ پرش می کنیم در غیر این صورت الگوریتم کدبرگردان پایان می یابد.

در معادله (۱۶) منظور از z ، تعداد مقادیر احتمالی که به انتهای رشته اطلاعات اضافه گردیده است که در مرحله ۱۰ کد کننده روش پیشنهادی محاسبه گردیده بودند و در مرحله ۱۴ کدکننده روش پیشنهادی به انتهای رشته بیت اطلاعات اضافه گردیدند.

- مرحله ۸: در این مرحله مقادیر دیگر تابع چگالی احتمال
- با استفاده از معادله (۱۳) محاسبه می گردند و بدین

مختلف در جدول (۱) نشان داده شده است. همان‌طور که در این جدول مشاهده می‌شود به دلیل زیاد بود میزان فشردگی رسته بیت ساخته شده، نرخ کدگذاری کانال ثانویه R_e' از نرخ کدگذاری کانال اصلی R_e کمتر شده است. به این معنا که تعداد بیتی که برای کد کردن کانال ثانویه R_e' به کار می‌رود بیشتر از تعداد بیتی است که برای کد کردن کانال اصلی R_e به کار رفته است. بنابراین، فریم‌های ارسالی میزان قابل توجهی نسبت به خطا کانال مقاوم می‌گردند.

پس از پیاده‌سازی روش پیشنهادی، نتایج به دست آمده با نتایج حاصل از روش‌های دیسایانایاکی، فاروق و الانگوان مقایسه می‌گردد. شکل‌های (۲۱ - ۲۰) میزان PSNR به دست آمده ناشی از به کارگیری روش پیشنهادی در مقایسه با روش الانگوان برای دو فایل فورمن و سوزی با نرخ ارسال ۱۲۸ کیلوبیت بر ثانیه را نشان می‌دهد. همچنین، شکل‌های (۲۴ - ۲۲)، میزان PSNR به دست آمده ناشی از به کارگیری روش پیشنهادی در مقایسه با روش دیسایانایاکی برای تمامی فایل‌های ویدیویی اخبار، فورمن و فوتبال با نرخ ارسال‌های ۶۴، ۱۴۴ و ۲۵۶ کیلوبیت بر ثانیه نمایش داده است. در انتها روش پیشنهادی با روش فاروق مقایسه گردیده است که در آن، شکل‌های (۲۷ - ۲۵)، میزان PSNR به دست آمده ناشی از به کارگیری روش پیشنهادی در مقایسه با روش فاروق را برای تمام ۱۲۳ فریم Foreman نشان می‌دهد. تنها تفاوت شکل‌های (۲۷ - ۲۵)، در نرخ تولید منبع می‌باشد. به این معنا که هر چه نرخ کدکننده منبع بیشتر شود، فریم‌ها نسبت به اعوجاج و نویز کانال قوی‌تر می‌گردند. همچنین در شکل‌های (۳۰ - ۲۸)، میزان PSNR برای تمام ۱۰۵ فریم Walk محاسبه گردیده و به همراه نتایج به دست آمده از روش فاروق رسم شده است. در انتها پس از انجام مقایسه بین روش پیشنهادی و چندین روش جدید این نتیجه حاصل می‌شود که، روش پیشنهادی توانسته است بدون افزایش نرخ بیت ارسالی کیفیت فریم‌های دریافتی را افزایش دهد که این امر در نمودارهای مربوط نشان داده شده است و می‌توان نمودارهای انتهایی را دلیلی بر برتری روش پیشنهادی با روش‌های دیگر دانست.

۸- ارزیابی روش پیشنهادی

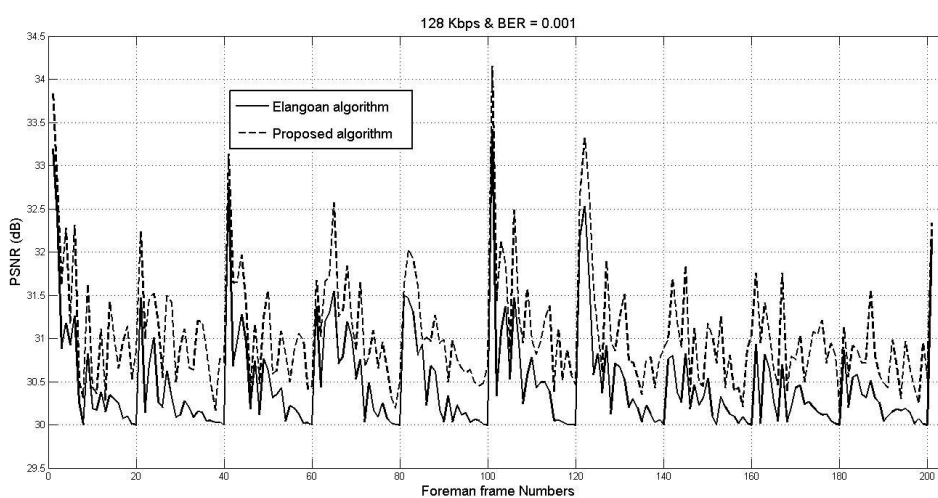
ابتدا به معرفی چندین پارامتر استفاده شده در شبیه‌سازی می‌پردازیم و سپس نتایج به دست آمده را با نتایج روش‌های دیسایانایاکی^۱، فاروق^۲ و الانگوان^۳ که به ترتیب در مراجع [۲۹-۲۷] آورده شده‌اند که از روش‌های جدید در خصوص کدگذاری همزمان منبع و کانال هستند، مقایسه می‌نماییم. کانال استفاده شده در این مقاله کانال باینری متقارن^۴ می‌باشد که خطای به وجود آمده در این کانال شامل نویز سفید گوسی^۵، فیدینگ با تابع چگالی احتمال رایلی^۶ است. در این مقاله از پنج قطعه ویدیو با نام‌های فورمن^۷، گردش‌پیاده^۸، اخبار^۹، سوزی^{۱۰} [۳۰] و فوتبال^{۱۱} [۳۱] استفاده شده است. به عنوان مثال ویژگی دو نمونه از فایل‌های ویدیویی را شرح می‌دهیم. برای دیدن ویژگی فایل‌های دیگر می‌توان به مرجع معرفی شده، مراجعه کرد. تعداد فریم‌های ویدیویی Foreman برابر با ۱۲۳ فریم است که از این تعداد ۳ فریم به صورت تصویر I کد می‌شوند و ۱۲۰ فریم دیگر به سه قسمت ۴۰ تایی که به صورت تصویر P کد می‌شوند و سپس ارسال می‌گردند. البته باید به این نکته اشاره نمود که ۳ فریمی که به صورت تصویر I کد شده‌اند در ابتدای هر قسمت ۴۰ فریمی که با تصویر P کد شده‌اند ارسال می‌گردند. تعداد فریم‌های Walk برابر با ۱۰۵ فریم است که از این تعداد ۵ فریم تصویر I و ۱۰۰ فریم تصویر P هستند که هر ۲۰ فریم تصویر P بین دو فریم تصویر I قرار دارد. سائز فریم‌ها برابر با 288×352 است. همچنین نرخ ارسال فریم برابر با ۲۵ فریم در ثانیه است.

این شبیه‌سازی را برای سه نرخ ارسال منبع پیاده‌سازی نموده‌ایم و با توجه به این که تعداد فریم ارسالی ثابت است (نرخ کدکننده منبع ثابت است)، هرچه نرخ ارسال بالاتر باشد، در نتیجه نرخ انکدر کانال بالاتر رفته و فریم‌های ویدیویی در مقابل نویز مقاوم‌تر می‌گردند. نرخ انکدر کانال R_e و همچنین نرخ انکدر کانال ثانویه R_e' برای دو فایل ویدیویی با نرخ‌های ارسال

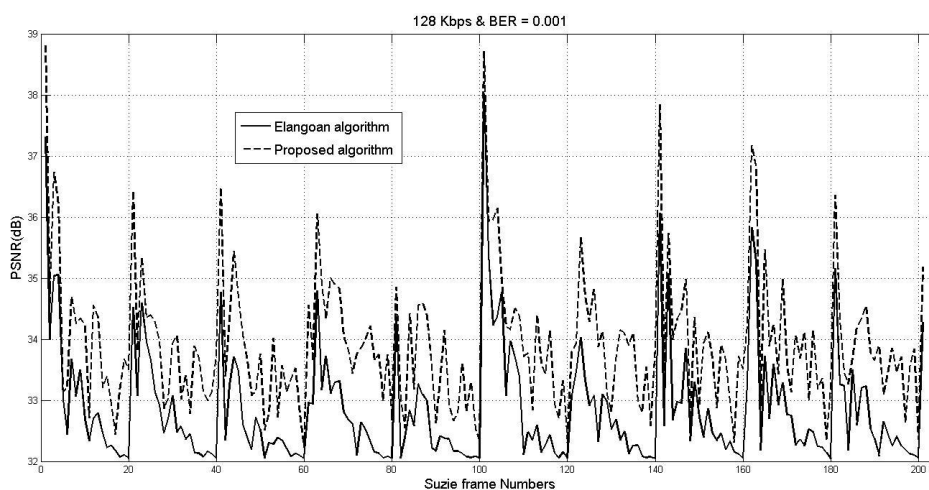
- 1- Maheshi B. Dissanayake
- 2- Muhammad Farooq
- 3- Premkumar Elangovan
- 4- Binary Symmetric Channel (BSC)
- 5- AWGN
- 6- Rayleigh fading PDF (Power Density Function)
- 7- Foreman Secuences
- 8- Walk Secuences
- 9- News Secuences
- 10- Suzie Secuences
- 11- Soccer Secuences

جدول (۱). نرخ انکدر کانال R_c و همچنین نرخ انکدر کانال ثانویه R'_c برای دو فایل ویدیویی Foreman و Walk با نرخ‌های ارسال مختلف

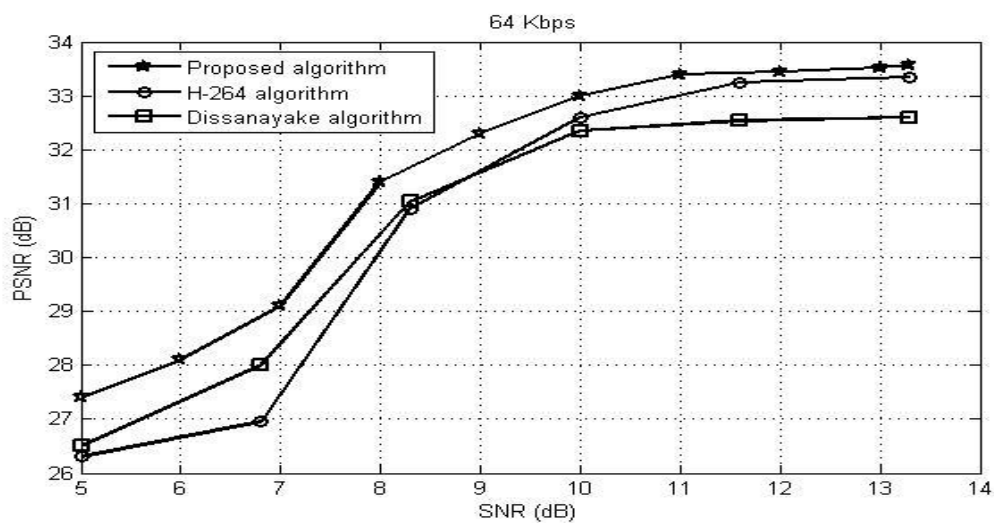
نرخ ارسال منبع	فایل ویدیویی ارسال شده	R_c	R'_c
۳۸۴Kbps	Foreman	۰/۴۶۶	۰/۱۰۷
۷۶۸Kbps	Foreman	۰/۲	۰/۰۵۴
۲Mbps	Foreman	۰/۰۶۴۵	۰/۰۱۲
۲۵۶Kbps	Walk	۰/۶	۰/۱۷۵
۵۱۲Kbps	Walk	۰/۳۳	۰/۰۸۴
۱/۵Mbps	Walk	۰/۱۲۹	۰/۰۲۶



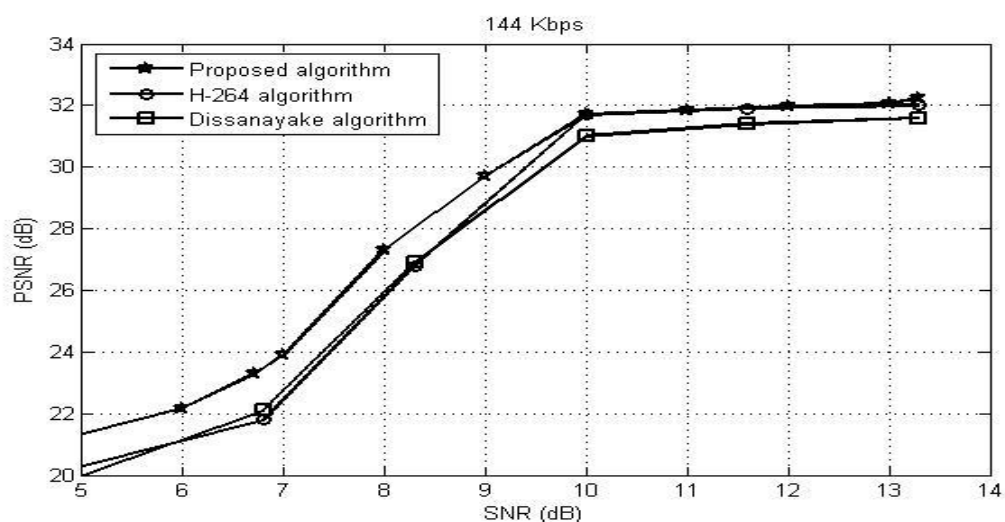
شکل (۲۰). مقایسه PSNR برای تمامی فریم‌های گردش فورمن روش پیشنهادی و الگوریتم الانگوان با مقدار 128 Kbps



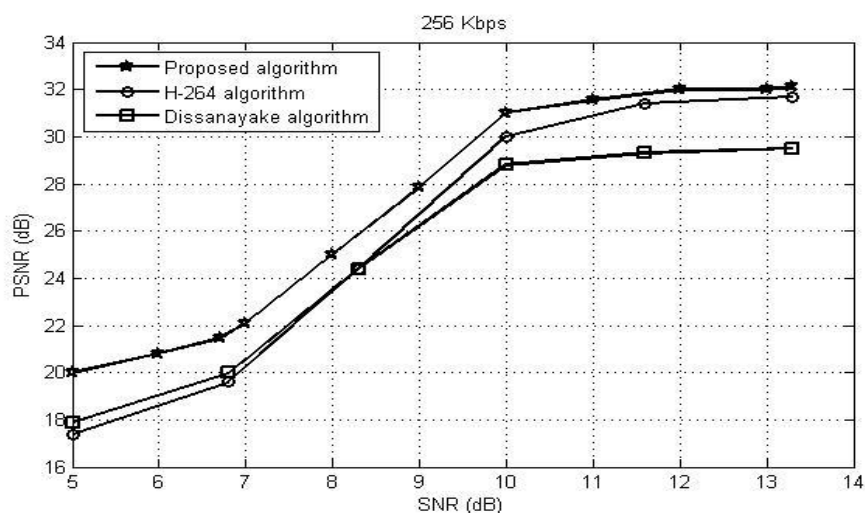
شکل (۲۱). مقایسه PSNR برای تمامی فریم‌های گردش سوزی روش پیشنهادی و الگوریتم الانگوان با مقدار 128 Kbps



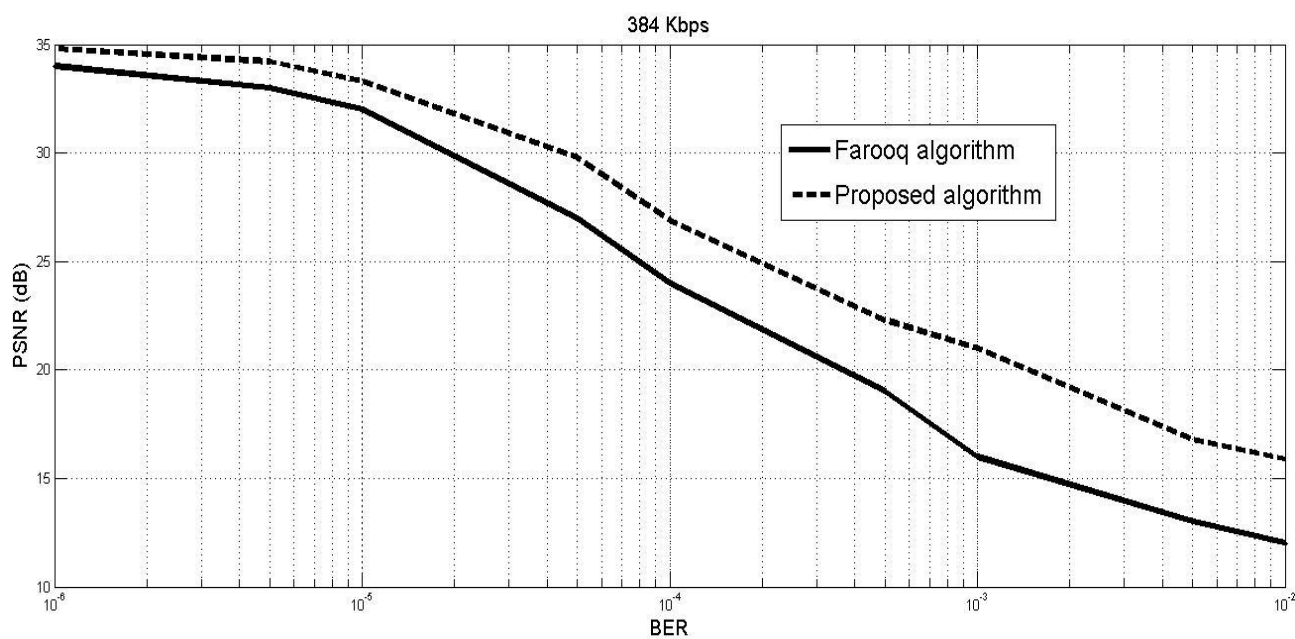
شکل (۲۲). مقایسه میانگین PSNR برای تمامی فریم‌های اخبار بین روش پیشنهادی و الگوریتم دیسایانایاکی با نرخ تولید کدکننده منبع با مقدار 64 Kbps



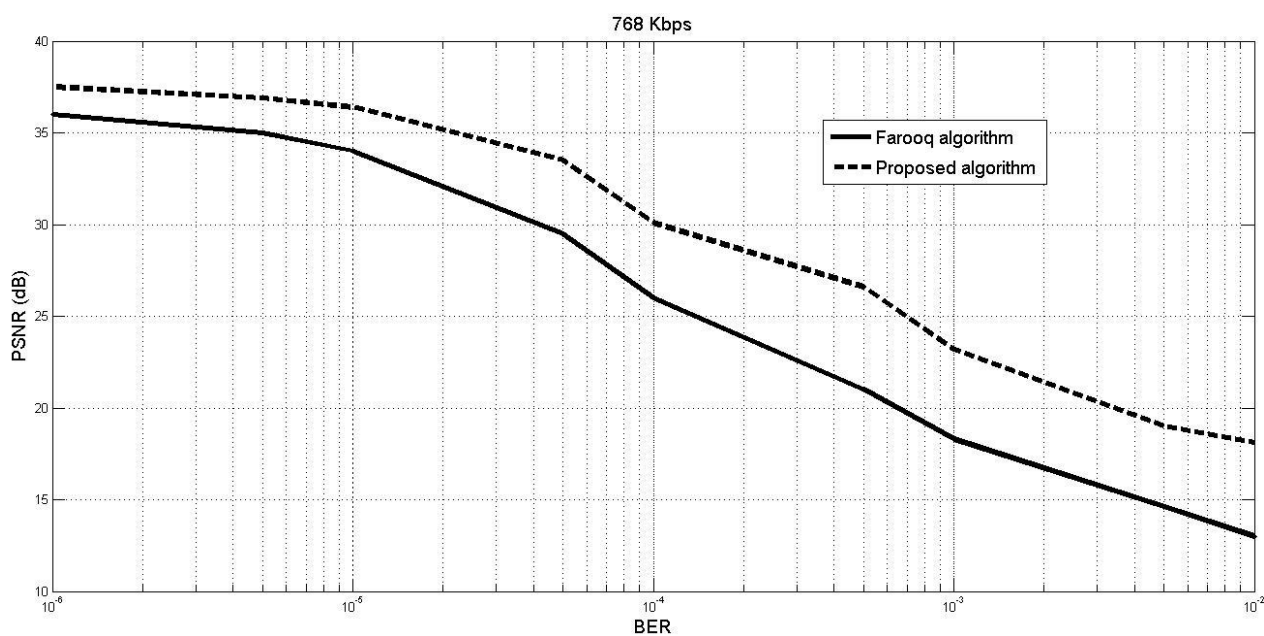
شکل (۲۳). مقایسه میانگین PSNR برای تمامی فریم‌های فورمن بین روش پیشنهادی و الگوریتم دیسایانایاکی با نرخ تولید کدکننده منبع با مقدار 144 Kbps



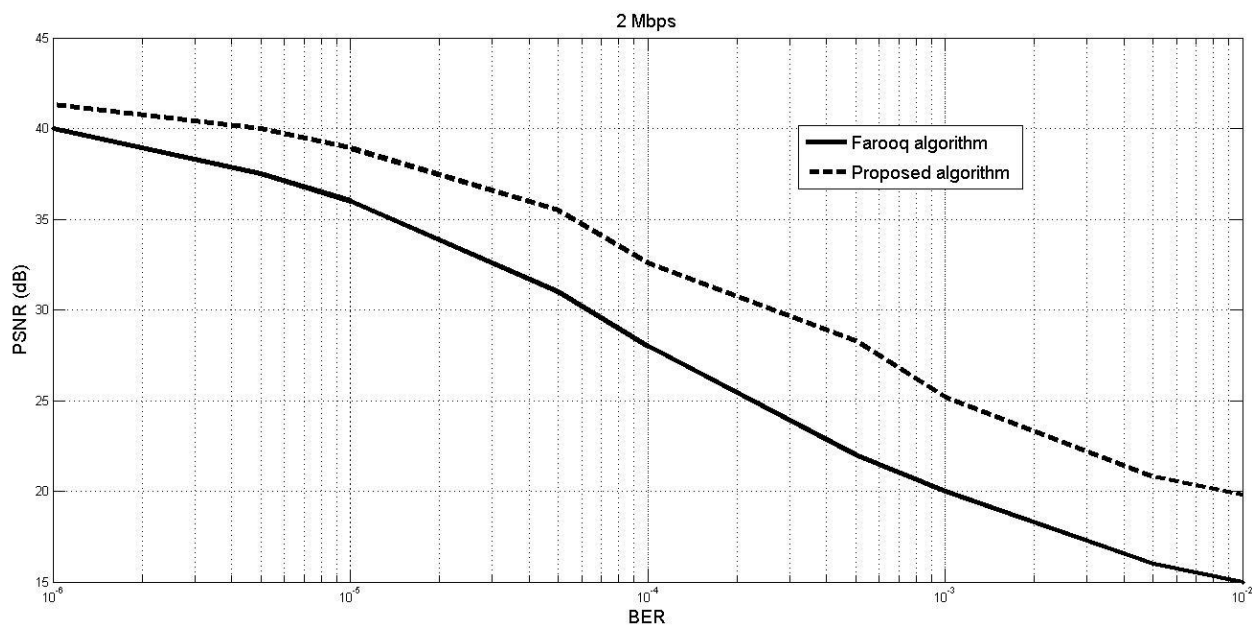
شکل (۲۴). مقایسه میانگین PSNR برای تمامی فریم‌های فوتبال بین روش پیشنهادی و الگوریتم دیسایانایاکی با نرخ تولید کدکننده منبع با مقدار 256 Kbps



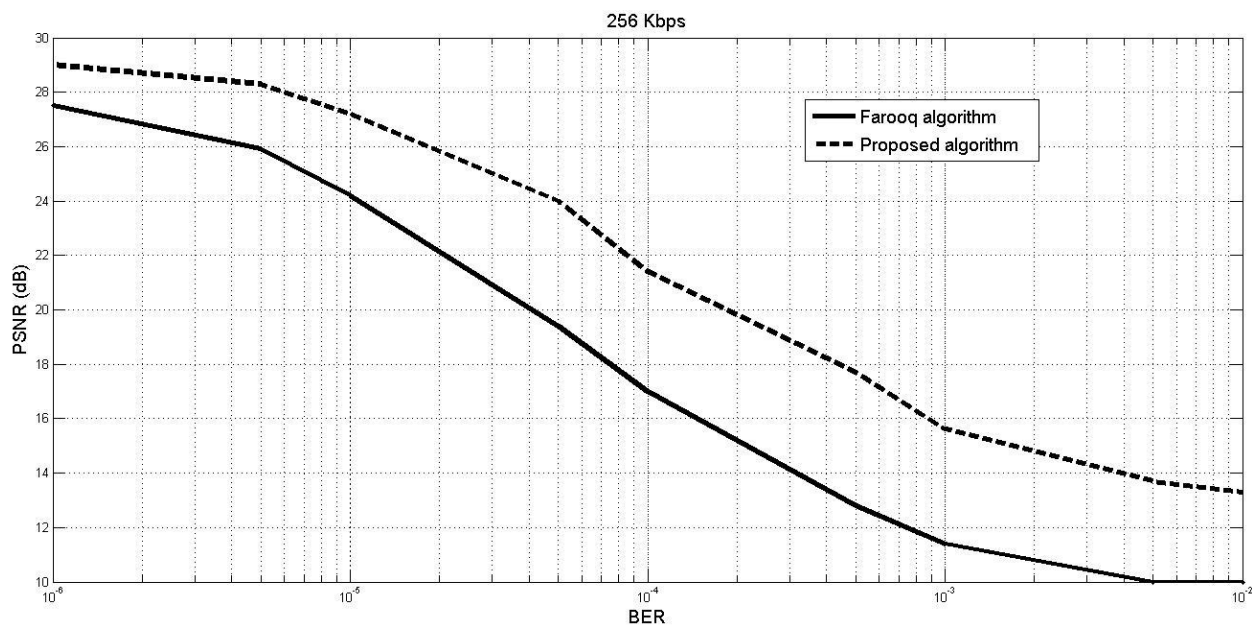
شکل (۲۵). مقایسه میانگین PSNR برای تمامی فریم‌های فورمن بین روش پیشنهادی و الگوریتم فاروق با نرخ تولید کدکننده منبع با مقدار 384 Kbps



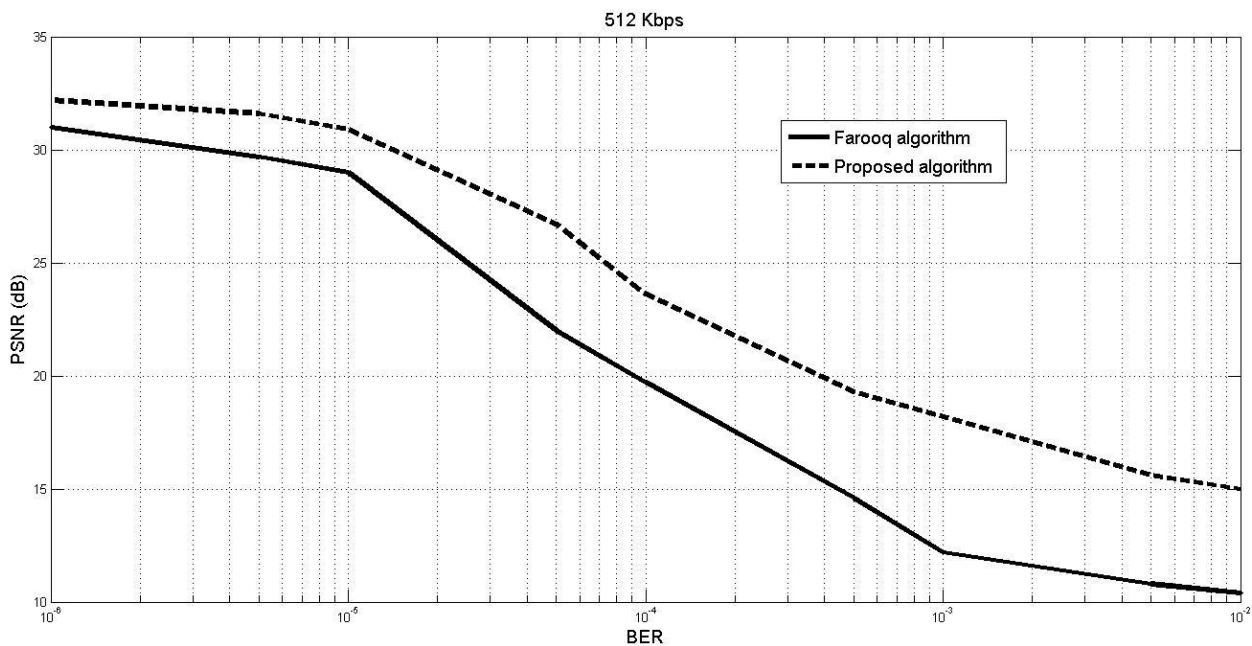
شکل (۲۶). مقایسه میانگین PSNR برای تمامی فریم‌های فورمن بین روش پیشنهادی و الگوریتم فاروق با نرخ تولید کدکننده منبع با مقدار 768 Kbps



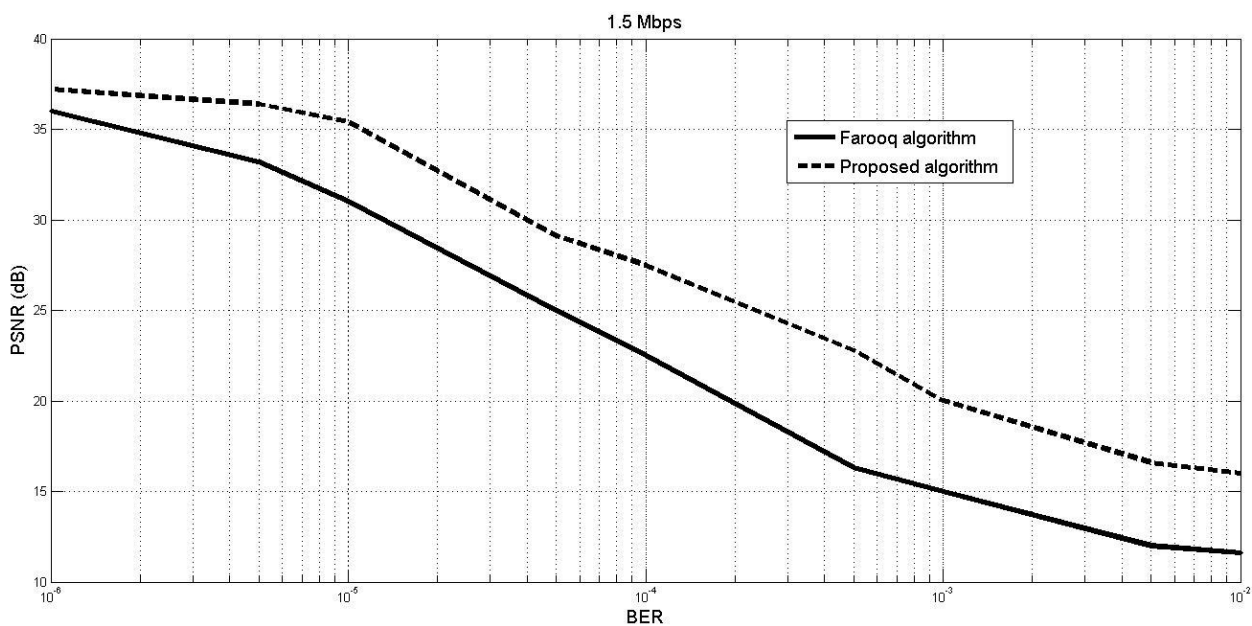
شکل (۲۷). مقایسه مقایسه PSNR برای تمامی فریم‌های فورمن بین روش پیشنهادی و الگوریتم فاروق با نرخ تولید کدکننده منبع با مقدار 2 Mbps



شکل (۲۸). مقایسه میانگین PSNR برای تمامی فریم‌های گردش پیاده بین روش پیشنهادی و الگوریتم فاروق با نرخ تولید کدکننده منبع با مقدار 256 Kbps



شکل (۲۹). مقایسه میانگین PSNR برای تمامی فریم‌های گردش پیاده بین روش پیشنهادی و الگوریتم فاروق با نرخ تولید کدکننده منبع با مقدار 512 Kbps



شکل (۳۰). مقایسه میانگین PSNR برای تمامی فریم‌های گردش پیاده بین روش پیشنهادی و الگوریتم فاروق با نرخ تولید کدکننده منبع با

مقدار 1.5 Mbps

۹- نتیجه‌گیری

در روش پیشنهادی PSNR فریم‌های ویدیویی در گیرنده، دارای مقادیر بیشتری نسبت به روش‌های ارائه‌شده اخیر می‌باشد. به این معنا که فریم‌های دریافتی با روش پیشنهادی دارای کیفیت بالاتری نسبت به آن روش‌ها هستند. در روش پیشنهادی به دلیل فشردگی بیت‌های به دست آمده از بلوک VLC و جایگزین کردن بیت‌های ساخته‌شده توسط کدکننده کانال ثانویه (که از تفاضل بین بیت‌های ساخته‌شده توسط بلوک VLC و بیت‌های ساخته‌شده پس از فشردگی به دست می‌آیند) توانستیم کیفیت فریم‌های ویدیویی دریافتی را نسبت به روش‌های جدید افزایش دهیم. روش پیشنهادی با افزایش دادن نرخ کدگذاری کانال و در نتیجه محافظت بیشتر از بیت‌های ارسالی، ضمن ثابت نگه داشتن نرخ کلی ارسال، توانسته است کیفیت فریم‌های ویدیویی دریافتی را بهبود بخشد.

۱۰- مراجع

- [13] Site: http://www.axis.com/products/video/about_networkvideo/compression.htm
- [14] Site: <http://www.axis.com/student/axisprize/prize1996/award1996.htm>
- [15] L. P. Kondi, F. Ishtiaq, and A. K. Katsaggelos, "Joint source-channel coding for motion-compensated dct-based snr scalable video," IEEE Trans. Image Process., vol. 11, no. 11, pp. 1043-1052, Sep. 2002.
- [16] MPEG Video Group, "MPEG-4 video verification model version 11.0," in ISO/IEC JTC1/SC29/WG11 MPEG98/N2172, Tokyo, Japan, Mar. 1998.
- [17] C. E. Shannon, "Communications in the presence of noise," Proc. IEEE, vol. 86, no. 2, pp. 447-458, February 1998.
- [18] B. Carlson, "Communication Systems," An Introduction to Signals and Noise in Electrical Communication, 3rd Edition, McGraw-Hill, New York, 1986.
- [19] C. E. Shannon, "A mathematical theory of communication," Bell Syst. Tech. J., vol. 27, pp. 379-423, pp. 623-656, July and October 1948.
- [20] V. Vemuri, "Artificial Neural Networks," Theoretical Concepts, Computer Society Press Technology Series, Computer Society of Institute of Electrical and Electronics Engineers, Washington, DC, 1988.
- [21] M. W. Gardner and S. R. Dorling, "Artificial Neural Networks (The Multilayer Perceptron)," A Review of Applications in Atmospheric Sciences, Atmos, Environ, 1998.
- [22] W. G. Cobourn, L. Dolcine, M. French, and M. C. Hubbard, "A Comparison of Nonlinear Regression and Neural Network Models for Ground-Level Ozone Forecasting," J. Air & Waste Manage, pp. 1999-2009.
- [23] M. Stone, "Cross-validated choice and 1 assessment of statistical predictors," Journal of the Royal Statistical Society, vol. 36, pp. 111-147, 1974.
- [24] K. Fukunaga, D. M. Hummels, "Leave-one-out procedure for nonparametric error estimate," IEEE transactions on pattern analysis and machine intelligence, vol. 11, pp. 421-430, 1989.
- [25] J. G. Proakis, "Digital Communications," McGraw Hill, Hardcover, 1995.
- [26] D. A. Huffman, "A method for the construction of minimum redundancy codes," Proc. IRE, vol. 40, pp. 1098-1101, 1952.
- [27] M. B. Dissanayake, "A Novel Error Robust Video Coding Concept Using Motion Vectors and Parity Bits," 7th IEEE International Conference on Industrial and Information Systems (ICIIS), pp. 1 - 6, 2012.
- [28] M. Farooq Sabir, R. W. Heath and A. Cornard Bovik, "Joint Source-Channel Distortion Modeling for MPEG-4 Video," IEEE Trans. Image Processing, vol. 18, no. 1, January. 2009.
- [29] P. Elangovan, "Motion Vector Smoothing Algorithm for Robust Wireless Multimedia Communications," 4th IEEE International Conference on Circuits and Systems for Communications (ICCSC), pp. 466-470, 2008.
- [30] http://see.xidian.edu.cn/vips/database_Video.html
- [31] Video Test Media; in <https://media.xiph.org/video/derf/>
- [1] M. Flierl and B. Girod, "Video Coding with Superimposed Motion-Compensated Signals," Kluwer Academic, ISBN 1-4020-7759-9, 2004.
- [2] J. C. Moreira and P. G. Farrell, "Essentials of Error-Control Coding," John Wiley & Sons, Ltd, 2006.
- [3] M. Bystrom and J. W. Modestino, "Combined source-channel coding schemes for video transmission over an additive white Gaussian noise channel," IEEE J. Sel. Areas Commun., vol. 18, no. 6, pp. 880-890, Jun. 2000.
- [4] G. Cheung and A. Zakhor, "Bit allocation for joint source/channel coding of scalable video," IEEE Trans. Image Process., vol. 9, no. 3, pp. 340-356, Mar. 2000.
- [5] L. P. Kondi, F. Ishtiaq, and A. K. Katsaggelos, "Joint source-channel coding for motion-compensated dct-based snr scalable video," IEEE Trans. Image Process., vol. 11, no. 11, pp. 1043-1052, Sep. 2002.
- [6] F. Zhai, Y. Eisenberg, T. Pappas, R. Berry, and A. Katsaggelos, "Rate-distortion optimized hybrid error control for real-time packetized video transmission," IEEE Trans. Image Process., vol. 15, no. 1, pp. 40-53, Jan. 2006.
- [7] ISO/IEC 11172-2- Standard, "Information technology Coding of moving pictures and associated audio for digital storage media - Part 2: Video," .
- [8] ISO/IEC 10918-6- Standard, "Information technology Digital compression and coding of continuous -tone still images: Application to printing systems," .
- [9] D. Salomon, "Data Compression," Third Edition, Springer, 2004.
- [10] F. Pereira and T. Ebrahimi, "The MPEG-4 Book," 1's Edition, Prentice Hall, July 2003.
- [11] I. E. G. Richardson, "H.264 and MPEG-4 Video Compression Video Coding for Next-generation Multimedia," New York: Wiley, 2003.
- [12] Y. Q. Shi and H. Sun, "Image and Video Compression for Multimedia Engineering- Fundamentals- Algorithms- and Standards," ISBN 0-8493-3491-8, 1999.

Quality enhancement of received video using secondary channel encoding in Joint Source and Channel Coding (JSCC)

P. Etezadifar, H. Farsi*, A. Naseri,
University of Birjand
(Received: 06/01/2015, Accepted: 01/09/2015)

ABSTRACT

Progress of technology in recent decade causes that video transmission via communication channels has met high demands. Therefore, several methods have been proposed to improve the quality of video under channel errors. The aim of this paper is to increase PSNR for synthesized video by increasing channel encoder rate but in constant transmission rate. This is achieved by using intelligent neural network and Huffman used in the MPEG standard to compress transmitted data significantly. Then, depending to the amount of compression by the proposed method, the compressed data is coded again using secondary channel encoder. The proposed method is able to increase channel coding rate without increasing the amount of information for each frame. This method provides more robustness for video frames against channel errors. The proposed method is tested for different source coding rates and several SNRs for channel and the obtained results are compared with state-of-the-art methods.

Keywords: video coding, variable bit rate, channel coding, joint source and channel coding.

* Corresponding Author Email: hfarsi@birjand.ac.ir