
**Università degli Studi di Padova**

Dipartimento di Ingegneria dell'informazione


*Corso di Laurea Magistrale in Ingegneria Informatica*

# METODO COGNITIVO NO-REFERENCE PER LA STIMA DELLA QUALITA' DEI VIDEO

**Tesi Magistrale**


*Laureando*

**Emanuele Giordano**

*Relatore*

**Prof. Michele Zorzi**

*Correlatore*

**Prof. Antonio Liotta**



20/04/2015

Anno Accademico 2014/2015

*A mia madre, a mio padre, a mio fratello e a mia sorella,*
*e anche questa è fatta....*

**Sommario**

La complessità delle rete di comunicazioni è in continuo aumento con l'aumentare dei diversi tipi di dispositivi che si affacciano direttamente a queste reti e le richieste di applicazioni o servizi multimediali da parte dei clienti finali aumentano a dismisura, tutto questo comporta dei doveri da parte dei fornitori cioè quello di garantire delle performance per soddisfare le numerose richieste e rispettare le aspettative del cliente. Come conseguenza, le valutazioni effettuate con le metriche del *Quality-of-Service (QoS)* falliscono nel cercare una combinazione delle proprietà tra la rete e le applicazioni, nonché la percezione dell'utente finale la cosiddetta *Quality of Experience (QoE)*. Esprimere la soddisfazione dell'utente dipende non solo dal giudizio personale, ma anche dalla aspettativa che l'utente ha nei confronti dell'applicazione, e quindi, le metriche soggettive sono i metodi più adatti ad effettuare questo tipo di valutazioni.

Tuttavia, non è possibile utilizzare questi test soggettivi quando è necessario un feedback in tempo reale. Le metriche oggettive *QoE (oQoE)* vengono utilizzate per questo scopo, ma cercare di realizzare metodi automatici per la valutazione del *QoE* non è cosa banale. Quando il media originale non è disponibile (es. servizi video basati sulla rete), è necessario utilizzare le metriche *oQoE No-Reference (NR)*, dove la valutazione viene effettuata con le sole informazioni ricevute dal client, quindi la stima della qualità viene effettuata lato-client.

In questo elaborato viene proposto un metodo NR per la stima della qualità dei video, basato su una funzione lineare per la stima del Blur e del rumore per ogni frame. Una *Artificial Neural Network* è stata utilizzata per rendere l'algoritmo dinamico ed adattarsi automaticamente ad ogni tipo di video tramite le caratteristiche del video stesso, impostando i pesi che rumore e blur hanno sul degrado della qualità. Mediante un emulatore di rete, è stata valutata l'accuratezza di questo metodo. I risultati mostrano una alta correlazione tra il metodo proposto e altri metodi *Full-Reference* considerati lo stato dell'arte, ma che richiedono il media originale per la stima della qualità.

Questo elaborato segue a grandi linee la seguente struttura:

- Il primo capitolo introduce il mondo dei servizi multimediali e la crescente domanda di questi servizi, con le relative problematiche, e perché l'importanza di metodi per la stima della qualità.

- Nel secondo capitolo viene dato uno scenario di quello che è lo stato dell'arte di queste metriche, vengono introdotti i concetti delle metriche oggettive, e ci si concentra sui metodi No-reference. Viene anche data una piccola introduzione generica sul Machine Learning
- Nel terzo capitolo viene descritto il metodo per la stima della qualità frame per frame, analizzando nel dettaglio tutto il suo funzionamento.
- Nel quarto capitolo viene descritta l'implementazione del metodo proposto e tutti gli strumenti utilizzati per la realizzazione.
- Infine, nel quinto capitolo vengono presentati i risultati ottenuti durante le sperimentazioni e le relative conclusioni.

# Indice

# Capitolo 1

# Introduzione

## 1.1 Servizi nel campo multimediale

I dati multimediali sono diventati dei componenti essenziali nei vari servizi offerti dalla rete Internet. L'esplosione di questi contenuti multimediali come ad esempio immagini e video ha aperto la strada a nuovi servizi come ad es. video sharing, IPTV, streaming video, servizi real-time audio e video ecc.
Tutto questo è stato possibile grazie all'avanzamento tecnologico, a nuove tecniche di compressione molto efficienti, che riescono ad eliminare l'informazione ridondante permettendo l'abbassamento della bitrate e quindi un uso minore della banda di trasmissione, cosa anche molto importante è che questi servizi multimediali possono funzionare, come qualsiasi altro servizio offerto da internet, come best-effort e quindi tramite un semplice protocollo IP, oppure con qualità di servizio garantita laddove sono richieste garanzie sulle specifiche di trasmissione a favore dell'utente (es. IPTV).
Un ulteriore fattore importante è stato lo sviluppo di nuovi dispositivi mobile come ad es. smartphone o tablet, che grazie alle loro elevate capacità di calcolo sono in grado di connettersi alla rete Internet e richiedere vari servizi multimediali come ad esempio video streaming, servizi in real-time ecc.
Il traffico globale dei dati mobile cresce a dismisura come il numero di connessione dei dispositivi mobile che come previsto dalla Cisco[1], il numero di connessioni mobile supera il numero di persone sulla terra. Nel 2013 il 53% del traffico dei dati mobile consiste in video e si aspetta una crescita al 67% nel 2018[1]. Con questa grande crescita di esposizione di immagini e video all'occhio umano, l'interesse di offrire una qualità dell'esperienza (QoE) sempre maggiore cresce naturalmente.

[1] http://www.cisco.com/

## 1.2 Stima della qualità

Stimare la qualità di un servizio, nel nostro caso di servizi multimediali, è sempre stato un problema in qualsiasi campo, a partire da un ambito industriale e quindi alla qualità di un prodotto per finire in un ambito informatico dove si cerca di offrire sempre più servizi o applicazioni ad alta qualità. Cercare di offrire servizi di qualità è molto importante per le aziende in quanto riescono ad ottenere vantaggi economici rispetto alla concorrenza. Per definizione di qualità possiamo intendere come il soddisfacimento delle aspettative che il cliente ha nei confronti nel servizio o del prodotto.
Il problema della valutazione della qualità nei servizi è costituito principalmente dalla determinazione del metodo di calcolo. Al fine di ottenere la stima della qualità nei servizi, sono state sviluppate varie tecniche che, sebbene abbiano la caratteristica comune di analizzare i giudizi del cliente, risultano differenti. Ancora più difficile è cercare di stimare la qualità di un video streaming, in quanto, offrendo un servizio utilizzando Internet, che già di per sé è una rete costituita con la politica del best-effort, i pacchetti relativi ai frame del video possono essere smarriti, ritardati o cancellati, introducendo rumore o artefatti al video, degradando cosi la qualità del servizio. Da qui si era cominciato un processo di raccolta di opinioni del cliente da parte del fornitore del servizio, ricevendo così un feedback molto importante sulla qualità, dove alla fine dell'utilizzo del servizio, che sia la visione di un film streaming o la video-chiamata tramite Internet, veniva chiesto al cliente la sua opinione in base all'esperienza da lui vissuta durante l'utilizzo del servizio, compilando un semplice modulo.
Ma la tendenza ora è quella di cercare di sviluppare dei metodi di stima della qualità che cercano di effettuare la misura in maniera correlata alle opinioni delle persone e automatica senza l'intervento dell'uomo, cosi da rendere disponibile un feedback in tempo reale al service provider e dare una maggiore garanzia del servizio. Le caratteristiche che questi metodi dovevano avere, era quella di essere leggero a livello computazionale, cercare di estrapolare importanti features per la stima della qualità, e utilizzare solo l'informazione ricevuta dal client, cioè senza avere a disposizione informazioni sul video sorgente, questi metodo rientrano nelle metriche oggettive **No-reference**, di cui si discuterà in dettaglio nel prossimo capitolo.

## 1.3 L'importanza

Negli anni recenti, c'è stato un crescente interesse nel sviluppare metodi per la stima della qualità, leggeri da poter funzionare in dispositivi mobile e

possibilmente che funzionassero in real-time, questo dovuto alla grandissima diffusione dell'uso di servizi multimediali in contesti di comunicazioni wireless e sistemi di telecomunicazioni. L'importanza e l'applicazione di questi metodi trovano impiego in diverse aree, come per gli operatori di rete e i fornitori di servizi che hanno un forte interesse di cercare di quantificare oggettivamente il livello di qualità fornita all'utilizzatore finale e ai nodi interni della rete, questi metodi forniranno i dati necessari per adottare le impostazioni ottimali di rete tale da rendere sicura la soddisfazione del cliente ed evitare il churn, l'abbandono del servizio.
Lo scopo di utilizzare questi metodi serve anche per poter stabilire dei contratti tra i provider e i clienti, i cosiddetti *service-level agreements (SLA)*[2], dove i provider garantiscono al cliente un certo livello di qualità, e quindi servono dei metodi per poter monitorare la qualità.
In generale, i metodi oggettivi chiamati **No-reference**, sono ben studiati per funzionare in quei contesti come le connessioni wireless che hanno frequenza limitata e introducono molto rumore, e quindi non si possono avere trasmissioni pulite per avere delle features del video originale non sporcate dal canale. Altra cosa questi metodi devono essere leggeri e quindi da poter funzionare in sistemi real-time e cercare di adattare la qualità basandosi sui feedback forniti dal metodo. Questo spiega l'importanza e il motivo del perché la tendenza è quella di sviluppare metriche NR che soddisfano tutte queste richieste. Negli ultimi anni sono stati sviluppati diversi metodi che prevedono l'uso di diversi operatori come per esempio l'utilizzo di machine learning, o lo studio dei pixel dal lato client, quindi *l'Image Processing*, l'utilizzo della *trasformata discreta di Fourier (DFT)* o della *trasformata discreta del coseno (DCT)*, quindi come si può capire c'è una grossa comunità che si sta muovendo per cercare di sviluppare uno standard vero e proprio in un contesto **No-reference**.

## 1.4 QoS vs QoE

Come già detto, le valutazioni effettuate con i parametri del *Quality-of-Service (QoS)* falliscono nel catturare la correlazione tra utente finale e applicazione. I router e i switch presenti nella rete Internet sono stati implementati per funzionare con la politica del best-effort, cioè quella politica di cercare di realizzare il proprio compito più velocemente possibile con le risorse di-

[2]strumenti contrattuali attraverso i quali si definiscono le metriche di servizio che devono essere rispettate da un fornitore di servizi (provider) nei confronti dei propri clienti/utenti. Di fatto, una volta stipulato il contratto, assumono il significato di obblighi contrattuali

sponibili in quel momento, senza dare garanzie sulla consegna del pacchetto o il corretto ordine d'arrivo dei pacchetti. Rinunciare a caratteristiche di controllo del traffico e della congestione garantisce d'altra parte dei vantaggi in termini di costi e prestazioni assolute, riuscendo comunque a mantenere una buona qualità se le capacità della rete eccede di buona misura le richieste effettive.
Viene fatta una distinzione tra i tipi di servizi, alcuni vengono detti elastici altri inelastici, i servizi inelastici sono quei servizi che hanno bisogno di un minimo garantito, parlando di risorse di rete come la banda o la bitrate, per poter funzionare es. video-streaming, videogiochi online ecc, i servizi elastici all'opposto si adattano a qualsiasi scenario di rete, ed in genere utilizzano il protocollo *TCP*[3] che garantisce la consegna dei pacchetti e l'ordine dei pacchetti, ma è un protocollo molto pesante e lento in quanto sovraccarica la rete e aumenta la ridondanza dei pacchetti. Il *QoS* garantisce per quei servizi che non usano protocolli come il *TCP*, le performance del servizo. Le valutazioni delle performance delle reti, tradizionalmente vengono fatte con i mezzi del *QoS*.
Tuttavia, l'incremento esponenziale delle interconnessioni di dispositivi ha causato una crescita della complessità della rete e quindi tutte le valutazione basate sul QoS non sono più sufficienti [2]. Fattori come jitter, pacchetti persi, latency o bitrate, rappresentano statisticamente il comportamento della rete, ma non possono valutare accuratamente come l'imprevedibilità della rete può disturbare la percezione dell'utente finale che utilizza questi servizi, cioè la *Quality-of-Experience (QoE)*.
Il *QoE* è definito come il degrado del piacere o la noia dell'utente riferito all'utilizzo di un'applicazione o di un servizio, quindi per calcolare il *QoE* devono essere prese in considerazioni ogni tipo di fattore che influenzi la percezione dell'utente come la flessibilità, la mobilità, la sicurezza, il costo ecc. Oltre a dipendere dall'utente il *QoE* sarà influenzato anche dal tipo di dispositivo di riproduzione, dall'ambiente di utilizzo (es. in auto o in casa), dalle aspettative dell'utente, dai contenuti e dalla sua importanza. A causa della sua essenza soggettiva, i giudici legittimi per la qualità visuale sono gli uomini, dove le loro opinioni possono essere ottenute tramite analisi soggettive [3] e mediante il *Mean Opinion Score (MOS)* si ha un risultato numerico del *QoE*, giudicando il jittering, blockiness e artefatti del video visualizzato. Si e poi cercato di sviluppare dei metodi oggettivi, come nel caso del *QoS*, per

[3]Transmission Control Protocol, è un protocollo di rete a pacchetto di livello di trasporto, appartenente alla suite di protocolli Internet, che si occupa di controllo di trasmissione ovvero rendere affidabile la comunicazione dati in rete tra mittente e destinatario. È definito nella RFC 793 e su di esso si appoggiano gran parte delle applicazioni della rete Internet.

cercare di predire il *MOS* e avere dei metodi più diretti e risultati immediati. Quindi grazie a metodi oggettivi o soggettivi, che spiegheremo nella prossima sezione, è possibile misurare il *QoE* e poter stipulare contratti *SLA* più sicuri ed evitare il pagamento delle penali da parte dei provider.
Il *QoS* è più adatto per il monitoraggio di dispositivi interni alla rete come server, router o analizzare il traffico di rete, quindi utilizzato per la risoluzioni di problemi all'interno la rete. *QoE* e *QoS* non sono mutualmente esclusivi, utilizzando parallelamente tutte e due le metriche si ha un monitoraggio completo del servizio.

## 1.5 Metriche di valutazione

La qualità delle immagini o dei video può essere degradata durante le fasi di cattura, compressione, trasmissione, riproduzione, e visualizzazione dovuto alla distorsione che ognuno di queste fasi può introdurre al media. Quindi esiste una necessità di sviluppare dei metodi per la stima della qualità delle immagini o dei video.
Queste metriche di valutazione si possono suddividere in due categorie, le metriche soggettive e le metriche oggettive. Le metriche soggettive cercano tramite dei test su persone fisiche di calcolare l'opinione comune sulla qualità di un video o di un immagine, le metriche oggettive sono dei modelli matematici che cercano di approssimare con accuratezza i risultati delle metriche soggettive, quindi di sviluppare dei metodi automatici, tramite lo sviluppo di algoritmi che analizzano alcune caratteristiche del video o delle immagini, e senza l'intervento dell'occhio umano riescono a dare un risultato correlato con i risultati soggettivi.

### 1.5.1 Metriche Soggettive

Per determinare la qualità dei video percepita dall'uomo, che sarebbe l'utilizzatore finale, occorrerebbe effettuare delle valutazioni soggettive, questi test soggettivi sottopongono un gruppo di partecipanti, di solito non esperti, a visualizzare delle immagini o dei video e ad esprimere un giudizio sulla qualità percepita.
Questi esperimenti soggettivi vengono effettuati in laboratori con ambienti controllati. Una attenta pianificazione viene effettuata su diversi fattori per questi metodi di valutazione, come la selezione del materiale da sottoporre ai soggetti, le condizioni di visualizzazione, la scala di valutazione, e il tempo di presentazione, tutti questi fattori devono essere decisi a priori per gli esperimenti soggettivi. Vengono fornite delle linee guida, da parte (ITU-R)

BT.500[4] da seguire per condurre questi test soggettivi ed ottenere validi risultati. I risultati di questi test vengono utilizzati per computare il *mean opinion score (MOS)*. Il *MOS* servirà come base per comparare i risultati di altre metriche sviluppate, le metriche oggettive.
Un grosso problema delle metriche soggettive è che, condurre gli esperimenti è un lavoro lungo, laborioso e costoso in quanto devono essere utilizzate delle strutture e macchine costose e la selezione dei soggetti per i test deve essere fatta con accuratezza, quindi la tendenza è quella di sviluppare nuove metriche oggettive che possibilmente abbiano una buona correlazione con i risultati delle metriche soggettive. Le metriche oggettive in generale analizzano il video o l'immagine e cercano di estrapolare delle features o artifatti e tramite questi stimare la qualità dei video.

### 1.5.2 Metriche Oggettive

I metodi oggettivi per la stima della qualità, possono avere due approcci, quello ingegneristico o quello psicofisico [5]. L'approccio psicofisico cerca di modellizzare il *human-visual-system (HVS)*[4] usando aspetti come sensitività del contrasto e dell'orientazione, selettività di frequenza, pattern spaziali e temporali, e percezione del colore. Queste metriche possono essere utilizzate su una grande varietà di tipi di degradazione dei video ma sono computazionalmente pesanti. L'approccio ingegneristico cerca di sviluppare delle metriche più semplici basandosi su l'estrazione e l'analisi di alcune features o artefatti sui video e possono essere utilizzati anche attributi relativi al *HVS* e quindi considerare pure aspetti psicofisici.
Dunque, il concetto base per questi tipi di metodi è quello di analizzare i contenuti del video e calcolare la distorsione piuttosto che creare complessi modelli visivi.
Un insieme di caratteristiche o parametri relativi alla qualità di un immagine o di un video vengono messe in comune tutte insieme per stabilire un metodo oggettivo per la qualità che può essere utilizzato per predire il *MOS*.
Vedremo nel prossimo capitolo che in base all'informazione disponibile riguardo il degrado del video originale, che può essere utilizzato per misurare la qualità, queste metriche possono essere suddivise in tre classi: **full-reference (FR)**, **reduced reference (RR)**, e **no-reference (NR)**.

[4]Si tratta di un modello che cerca di simulare il comportamento del sistema visivo umano è utilizzato nella Computer Vision

## 1.6 Contributi apportati

In questo elaborato viene proposto un algoritmo oggettivo per la stima della qualità video, che ricade nelle metriche no-reference. Questo lavoro è stato condotto presso la Eindhoven University of Technology (NL) ed è stato accettato per essere presentato alla conferenza sul *Quality of Multimedia Experience (QoMEX)*[47]. L'algoritmo è stato pensato effettuando uno studio approfondito su quello che è lo stato dell'arte attuale per quanto riguarda le metriche no-reference. E' stata focalizzata l'attenzione sulle metriche no-reference in quanto non hanno bisogno di informazioni relative ai contenuti del video originale per il calcolo della qualità, l'ideale per i servizi basati su video che utilizzano internet.
L'algoritmo lavora in tempo reale, stimando la qualità frame per frame, è possibile settare il numero di frame da elaborare, questo in base alla potenza di calcolo che possiede il dispositivo in cui è in esecuzione il software. L'algoritmo è stato scritto in **Java** è può funzionare su tutti i dispositivi che supportano la **Java Virtual Machine (JVM)**[5] tra cui anche i dispositivi mobile *Android*.
Per il calcolo della qualità vengono estratte due features molto importanti che consistono nel blur e nel rumore presenti nel frame, queste due features sono state considerate da MG. Choi, JH. Jung, JW. Jeon nel loro lavoro [6] come le più importanti ad influenzare la qualità delle immagini, con il loro metodo, adattato al caso dei video viene stimato il blur ed il noise, e tramite una funzione lineare viene calcolata la qualità. Ad ogni parametro relativo al blur ed al noise sono associati dei pesi, questi hanno il compito di influenzare l'effetto che noise e blur hanno sul degrado della qualità, in quanto, in base ai contenuti del video, blur e noise hanno effetti differenti.
Per rendere l'algoritmo dinamico sono state applicate tecniche di *Machine Learning*, ed è stata utilizzata una **Artificial Neural Network (ANN)** per il calcolo dei pesi relativi ai parametri di blur e noise. La **ANN** prendendo in ingresso dati relativi ai contenuti dei video, restituisce in output i pesi relativi al blur e noise, rendendo così l'algoritmo capace di adattarsi a qualunque tipo di video. Per riuscire a sviluppare quest'ultima parte appena descritta, sono stati fatti degli studi per cercare di capire che tipo di correlazione esiste tra il tipo di video e il degrado della qualità.
L'algoritmo si può dividere in due grandi fasi: la prima è la fase di **analisi del bitstream**, che serve a capire il tipo di video grazie alla caratterizzazione del video tramite tre importanti parametri: **Complexity**, **Motion**, **Bitra-**

[5]La java virtual machine è il componente della piattaforma Java che esegue i programmi tradotti in bytecode dopo una prima compilazione.

**te**. Tramite questi tre parametri la fase di analisi del bitstream determina le impostazioni dell'algoritmo relativi alla soglia del blur e i valori dei pesi sulle quantità di blur e noise, che rappresentano quanto l'effetto blur ed il noise degradano la qualità per il tipo di video in esecuzione. Grazie a questa fase l'algoritmo diventa dinamico, capace di adattarsi ad ogni tipo di video. Cercare di stimare i valori di soglia blur e pesi, univoci per tutti i video è impossibile in quanto blur e noise hanno effetti differenti in base al tipo di video; la seconda fase è la **stima della qualità del video**, con i dati forniti dalla fase precedente si procede al calcolo della qualità globale del video.
Per migliorare il tempo di esecuzione per la stima della qualità di un frame sono state adottate *tecniche di multithreading*, ogni frame da processare viene diviso in quattro macro-blocchi e per ogni macro-blocco viene eseguito un thread per la stima della qualità. L'algoritmo è stato testato in laboratorio grazie ad un emulatore di rete, simulando la percentuale di pacchetti persi per la trasmissione di un video tramite il protocollo *RTP*[6]. In tal modo, è stato possibile valutare gli effetti della rete (QoS) sulla qualità complessiva del servizio di video streaming (QoE). Una volta ottenuti i risultati questi sono stati confrontati con i risultati del **SSIM**, una metrica full-reference considerata al momento lo stato dell'arte. E' importante notare che **SSIM** viene usata puramente a scopo di benchmarking, ma in quanto tecnica *full-reference* non sarebbe utilizzabile in pratica per la valutazione di servizi mobile di video streaming.

[6]Real-time Transport Protocol è un protocollo di rete utilizzato per la trasmissione di audio e video su reti IP

# Capitolo 2

# Stato dell'Arte

In questo capitolo descriveremo quello che è lo stato attuale delle metriche di stima della qualità, quali di queste metriche viene considerato come standard, come vengono classificati, descriveremo le caratteristiche e i strumenti utilizzati per calcolare la qualità. In base alle loro caratteristiche e alla loro complessità computazionale troveranno impiego in vari contesti, cioè, se possono funzionare in real-time in un ambiente di network, se sono eseguibili anche in dispositivi mobile, o se hanno bisogno di informazioni aggiuntive, relative al video originale, per la stima della qualità.

## 2.1 Metodi Soggettivi

Come già accennato, i metodi soggettivi per la stima della qualità di una immagine o di un video sono quei metodi che prevedono l'utilizzo di un gruppo di persone esperte per la visualizzazione dell'immagine o del video e raccogliere le opinioni sulla qualità da loro percepita. Questi test vengono svolti in ambienti controllati, la causa è che molti fattori potrebbero influenzare la percezione visiva e quindi anche i risultati, fattori come la luminosità della stanza, il tipo di display, la distanza visiva, la risoluzione, il contrasto, la luminosità potrebbero condizionare la qualità percepita dai soggetti in test. Per rendere i test sicuri e ottenere dei risultati veritieri vengono seguite delle linee guida definite nel (ITU-R) BT.500[4]. Le prassi da seguire per effettuare i test si compone nel, selezionare le sequenze di video da testare, settare il sistema in base a cosa si vuole valutare nello specifico, scegliere un metodo di test per la raccolta delle opinioni, selezionare un numero sufficiente di soggetti, effettuare il test, e calcolare lo score delle opinioni. Uno dei metodi standardizzati sul come sottoporre il video agli esperti e di raccolta delle opinioni è il *DSIS - Double Stimulus Impairment Scale*: all'esperto viene

presentato il video originale senza nessuna imperfezione, poi lo stesso video viene compromesso e gli viene chiesto di votare il secondo video utilizzando una scala di riduzione di valore (da "svalutazioni sono impercettibili" a "svalutazioni sono molto fastidiose").
Condurre questi test in generale un lavoro dispendioso sia in termini economici sia dal punto di vista temporale. Per questo la tendenza è quella di realizzare dei metodi oggettivi che riescono a predire lo score dei metodi soggettivi (Sezione: 2.2).

### 2.1.1 MOS

Il Mean opinion score (MOS) è un tipo di test soggettivo che è stato utilizzato per decenni nel campo delle reti telefoniche per ottenere, da un punto di vista dell'utilizzatore, la qualità della rete. Con l'avvento dell'era della digitalizzazione, si è incominciato ad applicare questo metodo di valutazione anche nel campo della telefonia tramite la rete IP, servizio chiamato comunemente VoIP, dove la voce viene campionata quantizzata e trasferita tramite la rete come qualsiasi altro servizio che usa Internet, e quindi soggetta a degrado in base ai pacchetti persi, pacchetti in ritardo o pacchetti fuori ordine e così via. Com'è stato per la telefonia anche per i video c'è stato lo stesso tipo di tendenza, quella di fornire servizi video basati sulle reti IP come la IPTV, e quindi anch'esse soggette a degrado di qualità.
Per avere delle valutazioni soggettive riguardo al degrado della qualità che la rete provoca ai video viene applicato anche in questo caso il MOS. In base a quali fattori della qualità devono essere calcolati, il MOS può effettuare diversi tipi di stime della qualità. Il test si svolge sottoponendo il soggetto alla visualizzazione di un video o immagine e dopo di che gli verrà chiesto di rispondere ad alcune domande utilizzando *"absolute category rating" (ACR).* I test *ACR* utilizzano cinque categorie per esprimere il giudizio, ad ogni categoria viene associato uno score che va da 1 a 5, come mostrato in tabella 2.1. Dopo aver raccolto un numero sufficiente di voti, si procede al calcolo dello score del MOS calcolando la media di tutti i risultati ottenuti con l'*ACR.*

Esistono altri tipi di MOS, il *Degradation Mean Opinion Score (DMOS)* e il *Comparison Mean Opinion Score (CMOS)* entrambi definiti nel (ITU-T) P.800[7]. Il *DMOS* sottopone al valutatore la visualizzazione del video originale e successivamente il video distorto e attraverso il *"degradation category rating" (DCR)* il valutatore esprime un giudizio riguardo la distorsione percepita, un esempio di DCR per quanto riguarda la stima della qualità audio viene proposta nella tabella 2.2. Il CMOS propone sempre i due media ma in ordine casuale, i due vengono proposti al soggetto come, un media di riferimento e uno da valutare, il valutatore compara i due video ed esprime un

| Category | Score |
|---|---|
| Excellent | 5 |
| Good | 4 |
| Fair | 3 |
| Poor | 2 |
| Bad | 1 |

Tabella 2.1: ACR rating categories

| Category | Score |
|---|---|
| Degradation is inaudible | 5 |
| Degradation is audible but not annoying | 4 |
| Degradation is slightly annoying | 3 |
| Degradation is annoying | 2 |
| Degradation is very annoying | 1 |

Tabella 2.2: DCR rating categories

giudizio attraverso il *"comparison category rating" (CCR)*, tabella 2.3, considerando il media di riferimento. Come per il MOS, in entrambi i casi si procede a fare la media di tutti gli score ottenuti per avere una stima della qualità percepita.

## 2.2 Metodi Oggettivi

Sviluppare metriche oggettive significa cercare di creare metodi che analizzano il flusso d'informazione disponibile e riescono ad estrapolare alcune carat-

| Category | Score |
|---|---|
| Much Better | 3 |
| Better | 2 |
| Slightly Better | 1 |
| About the same | 0 |
| Slightly Worse | -1 |
| Worse | -2 |
| Much Worse | -3 |

Tabella 2.3: CCR rating categories

teristiche importanti e tramite queste stimano la qualità in termini di QoE. Un metodo oggettivo si può ritenere preciso ed accurato quando il risultato ottenuto ha una buona approssimazione comparata con i risultati delle metriche soggettive. La necessità di avere questi metodi nasce dal fatto che, come già detto, sostenere i test soggettivi è un lavoro lungo e laborioso sopratutto in termini di tempo, avere a disposizione questi metodi permetterebbe di avere dei risultati sulla qualità in maniera più diretta. Inoltre questi metodi trovano un importante impiego in contesti di servizi web multimediali, dove tenere in continuo monitoraggio la qualità del video dal lato client è un prezioso feedback per migliorare il servizio.
I metodi oggettivi, al fine di misurare la qualità percepita dall'utente, catturano i fattori che si reputano responsabili del degrado della qualità, e quindi influenzano la QoE dell'utente. Tra i fattori più comuni [8] possiamo trovare:

- ritardo iniziale, è l'intervallo di tempo che passa da quando l'utente lancia in esecuzione il video streaming a quando effettivamente il video è in esecuzione.

- il numero delle interruzioni che si verificano durante l'esecuzione del video a causa del buffer vuoto e l'utente deve attendere che il buffer si riempi parzialmente, questo evento influenza molto in negativo il QoE [9], un altro importante fattore che viene preso in considerazione è la durata delle interruzioni, che in base alla durata ha effetti differenti sul QoE dell'utente.

- la qualità del file video, la qualità di uno video stream è basata dal encoding rate. L'encoding rate è la media dei dati richiesti per eseguire un secondo di video, anche questo influenza il QoE [10], altre caratteristiche del video che vengono utilizzati per stimare la qualità sono il contrasto, l'effetto blur [11] e il blockiness [12].

- Lo scambio di bitrate, utilizzato nel *Dynamic Adaptive Streaming over HTTP (DASH)*[1], dove in base alle condizioni della rete la bitrate si adatta ad essa, frequenti cambiamenti di bitrate degradano la QoE dell'utente [13].

In base all'informazione che si necessita del video originale che si utilizza per stimare la qualità, oltre a quella del video distorto, le metriche oggettive vengono classificate in tre categorie, Full-reference, Reduced-reference e No-reference[14], che vedremo in dettaglio nella prossima sezione.

[1] noto anche come MPEG-DASH, è una tecnica streaming con bitrate variabile che consente lo streaming ad alta qualità di contenuti multimediali su Internet utilizzando i tradizionali server Web HTTP

### 2.2.1 Full-Reference, Reduced Reference, No-Reference

In dipendenza dall'informazione del degrado relativa al video originale, che si necessità per la stima della qualità i metodi oggettivi vengono divisi in metodi full-reference (FR), reduced-reference (RR), e no-reference (NR) [14], come segue:

- **Metodi FR**: Con questo tipo di approccio, l'intero video o immagine originale è disponibile come riferimento. Perciò, i metodi FR sono basati comparando l'immagine o video distorto con l'immagine o video originale. Queste metriche sono adatte per tradizionali trasmissioni broadcasting e sistemi televisivi. Alcune delle metriche più usate che ricadono nelle metriche FR sono *Peak Signal to Noise Ratio (PSNR)*, e *Structural Similarity (SSIM)*, che vedremo nella prossima sezione.

- **Metodi RR**: In questo caso, non è richiesto avere a disposizione l'intera informazione relativa all'immagine o video originale ma solo alcune caratteristiche rappresentative riguardo la struttura o altri tipi di caratteristiche relative all'immagine/video originale. Il confronto con le informazioni ridotte dell'immagine/video con l'informazione corrispondente dell'immagine/video distorto sarà l'input per i metodi RR.

- **Metodi NR**: Questa è la classe di metodi oggettivi di stima della qualità che non richiedono nessuna informazione aggiuntiva relativa all'immagine/video originale, ma basano la loro stima cercando artefatti nel dominio spaziale e quindi analizzando i pixel che compongono l'immagine/video. I metodi che utilizzano anche l'informazione contenuta nel bitstream del formato dell'immagine/video per la stima della qualità vengono detti metodi ibridi, in quanto utilizzano tecniche basate sui pixel e tecniche basate sul bitstream. I metodi NR sono quelli più indicati in contesti di servizi online, in quanto stimano la qualità solo grazie all'informazione ricevuta dal client.

Come si è già capito, a causa della richiesta parziale o totale di dati originali, le metriche FR e RR non sono adatte per funzionare in ambienti con canali di comunicazioni limitati. Inoltre, la loro complessità ed esecuzione temporale li rendono inadatti per il monitoraggio dei servizi multimediali on-line. Per colmare questa lacuna, appaiono le metriche NR, che non basandosi su confronti o misurazioni di fattori esterni ma solo dei dati ricevuti per modellare la QoE, risultano più veloci e quindi applicabili in contesti di servizi on-line.

### 2.2.2 PSNR

*Peak signal to Noise Ratio, PSNR* [15], è un metodo FR, che esprime il degrado dell'immagine/video con il rapporto tra la potenza massima possibile di un segnale, nel nostro caso l'immagine/video originale, e il segnale degradato dal rumore. Poiché molti segnali hanno un vasto range dinamico, il PSNR è generalmente espresso in termini di decibel e quindi rappresentato in scala logaritmica.
PSNR viene comunemente usato per misurare la qualità della ricostruzione dei codec con *compressione lossy*[2] (ad esempio, per la compressione delle immagini). Il segnale in questo caso sono i dati originali, e il rumore è l'errore introdotto dalla compressione.
Quando si confrontano i codec di compressione, il PSNR è un'approssimazione della percezione umana della qualità di ricostruzione. Bisogna essere estremamente attenti con il range di validità di questa metrica; è solo esclusivamente valida quando viene utilizzata per confrontare risultati di dati che utilizzano stessi codec e hanno gli stessi contenuti[16].
Il PSNR viene stimato attraverso *l'errore quadratico medio (MSE)*. Data un immagine $I$ senza rumore e $K$ immagine con distorsione, il MSE è definio come:

$$MSE = \frac{1}{n * m} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

Dove, $n$ e $m$ sono le dimensioni dell'immagine, $i$ e $j$ sono indici che ciclano ogni pixel dell'immagine, mentre $I(i,j)$ è il valore dell'intensità luminosa del pixel nella posizione $i$, $j$ dell'immagine $I$, e $K(i,j)$ è il valore del pixel nell'immagine $K$. Il PSNR (in dB) viene definito come:

$$PSNR = 10 * \log_{10}(\frac{MAX_I^2}{MSE})$$

$MAX_I$ è il valore massimo che il pixel può avere. Quando il pixel è rappresentato usando 8 bits, $MAX_I$ è 255, per immagini a colori nello spazio $RGB$ cambia solo il MSE che sarebbe la differenza dei valori al quadrato divisi per la grandezza dell'immagine e per tre. Alternativamente, l'immagine può essere convertita in un differente spazio di colore come per esempio il $TcbCr$ o *HSL* [17] [18]

[2] è una classe di algoritmi di compressione dati che porta alla perdita di parte dell'informazione originale durante la fase di compressione/decompressione dei dati che la rappresentano

### 2.2.3 SSIM

il *structural similarity (SSIM)* [19], è un metodo per misurare le similarità tra due immagini, di conseguenza c'è bisogno dell'intera immagine originale per misurare la qualità dell'immagine distorta, e quindi questo metodo ricade nelle metriche FR. SSIM è stato progettato per migliorare i tradizionali metodi come il PSNR che utilizza l'errore quadratico medio sul pixel dell'immagine, ed è stato provato che i risultati del SSIM sono ben correlati con i risultati delle metriche soggettive, quindi approssimano bene la percezione dell'occhio umano. Grazie alla sua accuratezza, il metodo SSIM è considerato lo stato dell'arte delle metriche FR, e per questo motivo è stato scelto per confrontare i risultati di questo elaborato.
La differenza principale con gli altri metodi è l'approccio, il metodo SSIM cerca di stimare l'errore percepito, in altre parole, l'immagine distorta viene considerata come i cambiamenti percepiti nella informazione strutturale. L'informazione strutturale è l'idea che i pixel hanno una forte interdipendenza specialmente quando sono spazialmente vicini. Questa dipendenza trasporta importanti informazioni riguardo la struttura degli oggetti che compongono la scena visuale.
La metrica SSIM è calcolata su varie finestre dell'immagine. La misura viene fatta tra due finestre $x$ e $y$ di grandezza uguale $N$x$N$ con la seguente formula:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

con:

- $\mu_x$ e $\mu_y$ media dei valori di $x$ e di $y$
- $\sigma_x^2$ e $\sigma_y^2$ varianza di $x$ e di $y$
- $\sigma_{xy}$ covarianza di $x$ e di $y$
- $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ due variabili per stabilizzare la divisione
- $L$ range dinamico dei valori dei pixel (tipicamente $2^{\#\, bits\, per\, pixel} - 1$)
- $k_1 = 0.01$ e $k_2 = 0.03$ di default

La valutazione del SSIM viene fatta solo con i valori luma[3]. Il risulato dell SSIM è un valore compreso tra -1 e 1.
Tipicamente il calcolo viene fatto su finestre di grandezza 8x8. La finestra

[3]rappresenta la luminosità di un'immagine (la parte "black-and-white" o acromatica dell'immagine)

può essere applica a tutta l'immagine pixel per pixel ma gli autori suggeriscono di applicare la finestra solo ad un possibile sottogruppo dell'immagine per ridurre la complessità computazionale. E' stato dimostrato da alcune ricerche [20], che il metodo SSIM non è veramente preciso, la qualità calcolata non è tanto correlata a quella percepita dall'occhio umano, ma tutto sommato è una buona approssimazione e per questa ragione è utilizzata da molti esperti per stimare la qualità di immagini e video e la degradazione dovuta a compressione e trasporto.

## 2.3 Metodi No-Reference per la stima della qualità di Immagini e Video

Negli anni recenti, c'è stato un crescente interesse nello sviluppo di metodi NR dovuto ad un enome diffusione dei servizi multimediali in contesti di comunicazioni wireless e sistemi di telecomunicazioni tradizionali. L'applicazione dei metodi NR trovano utilità in diverse aree, tra cui:

- fornitori di servizi che hanno un forte interesse di cercare di quantificare oggettivamente il livello di qualità fornita all'utilizzatore finale e ai nodi interni della rete, questi metodi forniranno i dati necessari per adottare le impostazioni ottimali di rete tale da rendere sicura la soddisfazione del cliente ed evitare il churn, l'abbandono del servizio.

- Il coinvolgimento di più parti tra i fornitori del servizio e gli utenti finali dà luogo a stabilire contratti di servizio (SLA), in base al quale viene concordato la garanzia di un certo livello di qualità. A questo proposito, i metodi NR sono la scelta più adatta per monitorare la qualità del servizio in sistemi live.

- In generale, i metodi NR sono adatti per eseguire in tempo reale la valutazione della qualità oggettiva in contesti di trasmissioni dove le risorse sono limitate, come per esempio lo spettro di frequenza nelle comunicazioni wireless. In tali casi, i metodi RR hanno una applicazione molto limitata, in quanto servirebbe un secondo tipo di canale trasmissivo dove trasmettere le caratteristiche necessarie per la stima del video originale.

- Nei servizi di comunicazione in real-time e nei servizi streaming vengono richieste adattamenti di qualità con i metodi NR per una raccolta statistica della qualità erogata.

In accordo con il framework introdotto in [21] per la stima della qualità in contesti NR, vengono presentati tre fasi fondamentali per il calcolo. Queste tre fasi sono: le misure di quantità fisiche rilevanti ai fini della qualità visiva, chiamati più comunemente features; la seconda fase consiste nel mettere insieme le misure effettuate a prescindere se questi sono misure temporali e/o spaziali; ed infine la mappatura dei dati raccolti al fine di stimare la qualità.

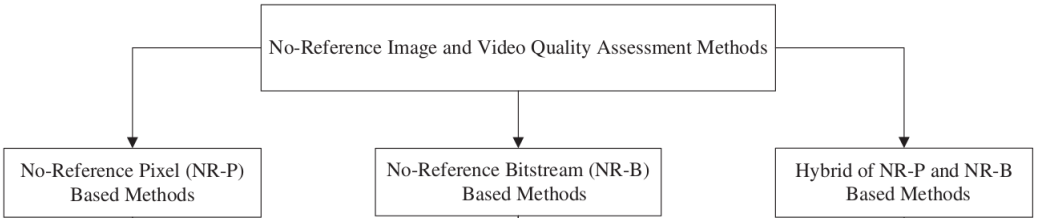


Figura 2.1: Classificazione metodi NR [2.3]

Utilizzando la classificazione proposta da Shahid e Rossholm [22], figura 2.1, possiamo suddividere i metodi NR in base al tipo di analisi che effettuano per la stima della qualità, questi possono essere, NR pixel (NR-P) se effettuano la loro stima solo su features basate sui pixel, e quindi applicano tecniche di computer vision, NR bitstream (NR-B) che utilizzano dati provenienti solo dal codec bitstream, e metodi ibridi che utilizzano informazioni sia dai pixel che dal bitstream.

### 2.3.1 Metodi No-Reference basati sui pixel (NR-P)

L'effetto blur, il blocking, ed il rumore sono considerati da molti come gli artefatti nel dominio spaziale più presenti nelle immagine/video dovute alle compressioni lossy, compressioni con perdita d'informazione [23]. Anche attraverso la trasmissione di immagine/video per via di reti a pacchetto senza garanzie sui pacchetti, possono venire introdotti questi artefatti.
Esistono altri tipi di artefatti che possono essere presenti come distorsione di immagine/video, ma per questi tipi di metodi che devono essere leggeri a livello computazionale, non possono venire presi in considerazioni tutti i possibili artefatti, in quanto questi metodi lavorano con l'esaminare pixel per pixel l'intera immagine/video, quindi si sceglie di selezionare alcuni artefatti considerati i più importanti. Una volta trovati gli artefatti, questi possono essere combinati per la stima della qualità percepita. Come mostrato in figura 2.2, un immagine o un video viene processato per estrarre le caratteristiche rilevanti per i differenti artefatti, dopo di che, un adatto meccanismo

di combinazioni di questi risultati viene utilizzato per la misura dei differenti artefatti, e si procede con la stima della qualità percepita. I metodi

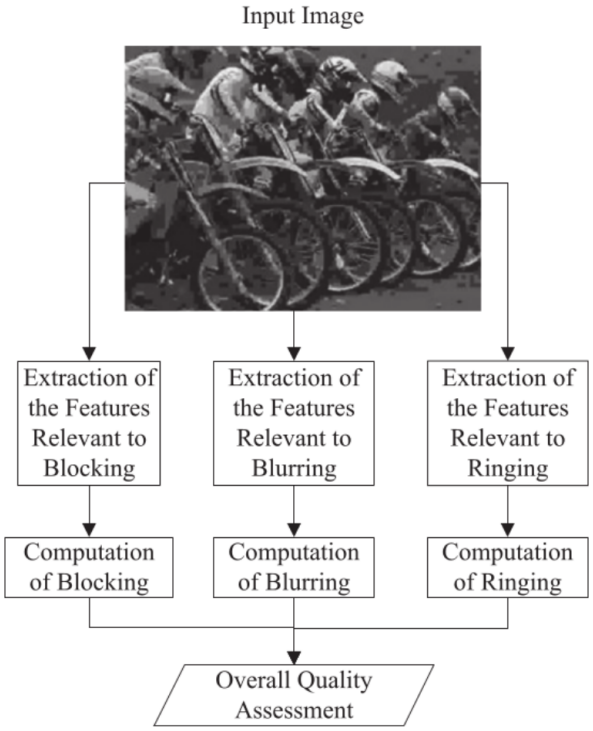


Figura 2.2: Schema base per metodi NR-P per la stima della qualità combinando diversi artefatti [2.3.1]

NR-P sono più indicati per la misura della qualità nelle immagine, in quanto non prendono in considerazione fattori temporali o informazioni di codifica molto importati nella stima della qualità dei video. Altra considerazione, è che questi metodi sono molto pesanti a livello computazionale e quindi poco adatti a funzionare in real-time, ma con l'avanzamento tecnologico e con un approccio più ingegneristico si riesce ad adattare questi metodi in contesti di servizi real-time ottenendo anche buoni risultati.

#### 2.3.1.1 blur

Winkler definisce l'effetto blur come un artefatto che appare come perdita dei dettagli spaziali e come una riduzione della nitidezza dei bordi[24]. Le ragioni del verificarsi dell'effetto blur possono essere molteplici, in origine nella acquisizione, trasformazione, o nella compressione [25]. In altri termini si può vedere l'effetto blur come una taglio delle alte frequenze trasformando l'immagine dal dominio spaziale a quello delle frequenze, il taglio delle alte

frequenze comporta l'eliminazione dei dettagli dell'immagine.

Uno schema di base NR per il calcolo del Blur viene riportato con il suo

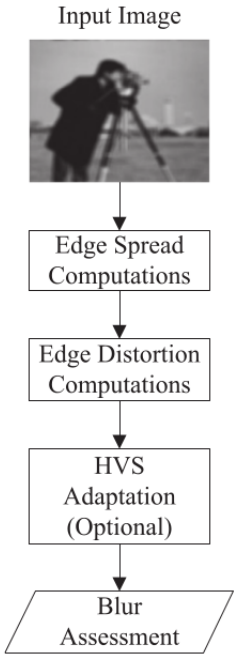


Figura 2.3: Schema base per metodi NR-P per la stima del blur [2.3.1.1]

diagramma di flusso nella figura 2.3. In molti metodi NR che utilizzano il Blur per stimare la qualità visiva, cominciano la computazione misurando la diffusione dei pixel presenti sui edge dell'immagine. Solitamente, si utilizzano tecniche di computer vision i cosiddetti *edge detector* quali **Sobel** e/o **Canny** per la rilevazione degli edge dell'immagine. Il passo successivo è in genere il calcolo della distorsione dei valori sui edge per cercare di stimare il blur complessivo. Alcuni metodi, tuttavia, fanno uso di un adeguamento HVS per la distorsione del valore dei edge per classificare se la distorsione è percettibile all'occhio umano oppure no. Questa è la tecnica più comunemente usata, ma molti altri metodi utilizzano diverse tecniche completamente diverse, come l'uso della *trasformata discreta di Fourier (DFT)* e procedere nell'analisi delle alte frequenze o utilizzano la *trasformata discreta del coseno (DCT)* e realizzare un istogramma con i coefficienti presenti nelle codifiche MPEG e JPEG ed analizzare i coefficienti nulli, oppure tecniche che integrano al metodo base tecniche di *machine learning* come l'uso di *artificial neural network* per classificare se un immagine è colpita da blur o no.

#### 2.3.1.2 noise

Il rumore spaziale è anche uno dei principali componenti di degrado della qualità visiva di immagine e video. La gran parte di tipi di rumori spaziali che si verificano sono il rumore salt and pepper, rumore di quantizzazione, rumore gaussiano e rumore speckle. Molti di questi rumori sono considerati componenti additivi, come per esempio il rumore gaussiano, ma in alcune situazioni i componenti del rumore sono moltiplicativi, come per esempio nel rumore speckle [26]. Il rumore può essere introdotto durante le fasi di acquisizione di immagini/video, registrazione, elaborazione, e trasmissione [27].
Simili agli approcci utilizzati per la stima del blur, il calcolo del rumore dipende dall'estrazione di alcune features che sono affetti da rumore. La figura 2.4 mostra lo schema di base di un approccio basato su blocchi per la stima del rumore, dove un'immagine è divisa in smooth areas. Una alta variazione rispetto ad una certa soglia all'interno di queste aree dà una stima del rumore. Un approccio sempre basato sui blocchi proposto in [28] utilizza analisi

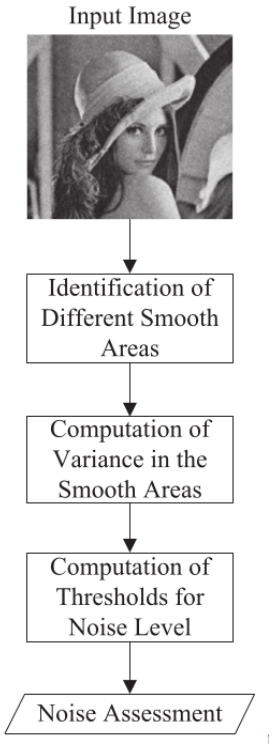


Figura 2.4: Schema base per metodi NR-P per la stima del rumore [2.3.1.2]

statistiche di un istogramma della varianza di un segnale locale per calcolare una stima della varianza del rumore dell'immagine. Tuttavia, questo metodo

ha un'alta complessità computazionale causa della sua elaborazione iterativa, e il metodo [27] semplifica questa tecnica prendendo in considerazione la struttura dell'immagine. Il metodo utilizza un filtro passa-alto per determinare l'omogeneità dei blocchi oltre che all'utilizzo della varianza del rumore medio.

#### 2.3.1.3 blocking

Il blocking è un artefatto che si manifesta come una discontinuità tra blocchi adiacenti in immagini o frame video [29]. È un tipo di degrado che si verifica dopo l'utilizzo di processi basato sui blocchi e tecniche di compressione ad alto coefficiente di compressione. In queste tecniche la trasformazione è di solito seguita dalla quantizzazione di ogni blocco che può causare nelle immagini o frame ricostruiti ad avere i margini dei blocchi incoerenti tra loro. Il Blockiness può essere stimato in una regione dell'immagine, in generale, calcolando la differenza tra blocchi vicini e la quantità di luminosità intorno a tali blocchi come mostrato nella figura 2.5. Dopo che il valore di blockiness è stato determinato in una certa regione, è importante valutare se sia significativo per la percezione umana o no tenendo conto anche degli effetti di mascheramento. In questo modo, possono essere calcolate alcune features che rappresentano l'input del HVS. In generale, la percezione del blocking e affetta da vari fattori, tra cui la forza blockiness (cioè, la differenza tra blocchi adiacenti), la luminosità locale attorno ai blocchi, e alla texture locale presente nell'immagine.

### 2.3.2 Metodi No-Reference basati sul Bitstream (NR-B)

Una stima della qualità di un video codificato può essere fatta analizzando il flusso di bit codificato per avere facilmente a disposizione alcune features come, parametri di codifica e qualità dei servizi di rete tutti parametri relativi al (QoS). I metodi che adottano l'uso dei dati provenienti dal bitstream per la stima della qualità non hanno un elevata complessità computazionale in quanto non hanno bisogno dell'elaborazione dei video completi, quindi non è richiesta una decodifica completa del video in input. Un altro vantaggio per questi tipi di metodi è l'uso di informazioni prontamente disponibili dal bitstream che sono rilevanti per la stima della qualità, ad esempio, i motion vector, la modalità di codifica e i valori dei parametri quantizzati. Tuttavia, questi metodi sono strettamente dipendenti dal tipo di codifica utilizzato, diversi tipi di codifica hanno un formato del bitstream differente. Esiste però una vasta gamma di caratteristiche rilevanti, ai fini della qualità, che

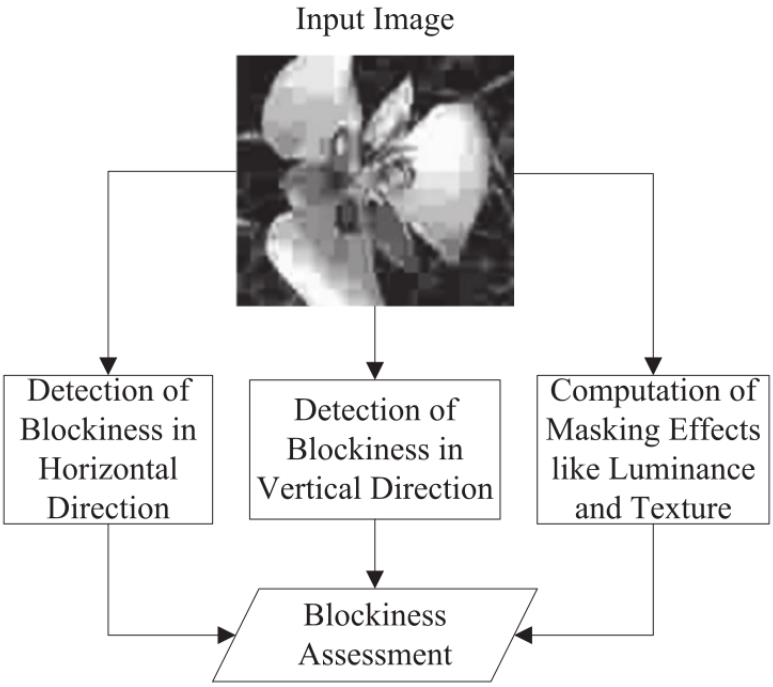


Figura 2.5: Schema base per metodi NR-P per la stima del blocking [2.3.1.3]

possono essere estratti dalla parziale decodifica o da una prima analisi dei dati presenti nel bitstream. La prestazione di tali metodi dipende in modo significativo dal livello di accesso al bitstream [30]. Viene riportato in figura 2.6 un diagramma a blocchi generico dei metodi basati sul bitstream. Questi metodi vengono divisi in tre categorie, in base al livello di informazioni che utilizzano per il calcolo della qualità, in accordo con gli standard definiti dal International Telecommunication Union (ITU-T), [31]. Questi tre livelli sono, il *parametric planning model*, *parametric packet-layer model*, e il *bitstream layer model*. Nei primi due modelli, vengono estratte caratteristiche estrinseche del video che sono di natura parametrica come bitrate, frame rate, e la percentuale dei pacchetti persi. Nel bitstream layer model si ha un accesso dettagliato al payload e a caratteristiche intrinseche legate al video, come la modalità di codifica, i parametri di quantizzazione, e i coefficienti DCT. La standardizzazione di questi modelli è valida anche per quei metodi progettati per la stima della qualità audio ma la nostra discussione è limitata a solo la qualità video.

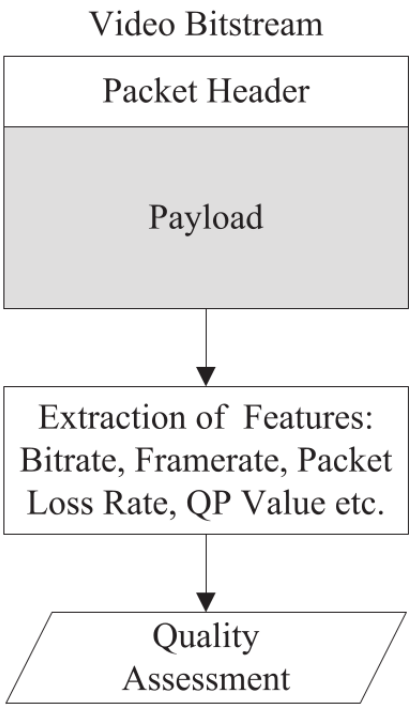


Figura 2.6: Schema base per metodi NR-B per la stima della qualità basandosi solo su features del bitstream [2.3.2]

#### 2.3.2.1 parametric planning model

Il parametric planning model ha una bassa complessità in quanto non accede al bitstream e utilizza bitrate, tipo di codec, e tasso di perdita dei pacchetti per effettuare un stima approssimativa della qualità video. Un modello di predizione della qualità, per video codificati in MPEG-2 e H.264/AVC per servizi di IPTV è presentato in [32]. Il modello prende alcuni parametri relativi alla codifica dei dati, le informazioni dei pacchetti e informazioni del cliente per valutare la qualità complessiva. In riferimento [33], un modello parametrico viene proposto basandosi su un semplice metodo di stima del MSE che si verifica a causa della perdita dei pacchetti. Gli autori utilizzano una relazione tra la lunghezza media del motion vector e il MSE e questa relazione dà una giusta stima del reale MSE.

#### 2.3.2.2 parametric packet-layer

Il parametric packet-layer ha accesso ai header dei pacchetti del bitstream e può estrarre un limitato insieme di parametri compresi bitrate, livelli di

frame, frame rate, tipo di frame, e il tasso di perdita dei pacchetti. I metodi basati sul parametric packet-layer sono noti anche come metodi basati sul QoS. Il ITU-T riconosce questi metodi come dei modelli parametrici non intrusivi per la valutazione delle performance dei servizi multimediali. Il metodo di stima della qualità visiva proposto in [34] presenta un approccio in cui non è richiesto decodificare il video ad ogni livello, questo è utile in quelle situazioni in cui il video viene codificato, in certi livelli, con l'uso della crittografia. Dato l'osservazione dell'errore che è più pesante quando c'è meno movimento nel video, viene richiesta una stima del moto dinamico per un particolare video, necessaria per valutare l'efficacia della strategia di occultamento dell'errore. In questo metodo, il rapporto tra la media della dimensione del B frame (bi-predittivo coded) con la media delle dimensioni di tutti i frame viene confrontata con un soglia predeterminata per regolare il valore dello score della qualità video. I risultati ottenuti dalla efficacia di errore vengono raffinati regolando i valori secondo l'importanza della regione in cui l'errore si è verificato.

#### 2.3.2.3 bitstream layer model

Nei metodi basati sul bitstream-layer, questi hanno accesso alla maggior parte dei dati che possono essere utilizzati per la stima della qualità video. Tali metodi utilizzano parametri del bitstream non intrusivi per la stima della qualità dei servizi multimediali. In questa modalità, è possibile fare qualsiasi tipo di analisi del bitstream tranne l'utilizzo dei dati pixel. Le informazioni in input includono parametri estratti dagli header dei pacchetti e dal payload. Oltre ai parametri inclusi nei modelli parametrici, questo tipo di modello utilizza anche i QP (parametri quantizzati), i coefficienti DCT del video codificato, e informazioni sui pixel. Ciò rende il modello comparativamente più complesso ma generalmente offre prestazioni migliori. Una soluzione a bassa complessità di stima della qualità basata sull'estrazione di parametri dal bitstream viene presentata in [35]. Le features utilizzate sono principalmente relative ai parametri di codifica e sono prese dal livello di sequenza. La bassa complessità è stata possibile grazie all'utilizzo di una semplice sistema di regressione multilineare per la costruzione delle relazioni tra i parametri e i valori di qualità.

### 2.3.3 Metodi No-Reference Ibridi

Esistono metodi no-reference di stima di qualità visiva che uniscono alcune caratteristiche del bitstream codificato e alcune statistiche da parte del media decodificato. Questi tipi di metodi ereditano la semplicità computazionale

degli approcci basati sul bitstream e la precisione della stima della qualità ottenuta aggiungendo in input gli approcci basati sui pixel. Pertanto, tali metodi possono evitare alcune delle difficoltà dei metodi basati sui pixel o basati sul bitstream [15]. Un esempio è la fusione di artefatti visivi come il

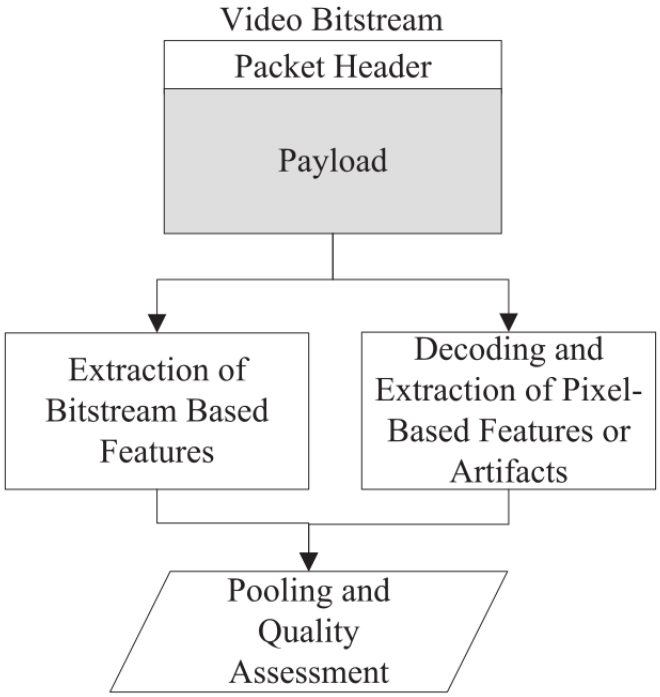


Figura 2.7: Schema base per metodi ibridi per la stima della qualità basandosi su features del bitstream e del dominio dei pixel [2.3.3]

blur e il blocking con i parametri derivati dal motion vectors per la costruzione di un metodo di stima ibrido. In figura 2.7 viene fornita una panoramica della metodologia utilizzata per questi tipi di metodi. Essenzialmente, la scelta delle caratteristiche da estrarre dal bitstream o dal dominio dei pixel dipende dal requisito di progettazione richiesto per il metodo, la disponibilità di un certo tipo di dati per la stima della qualità, e dallo schema di codifica. Questi metodi vengono divisi in due categorie, in base al tipo di analisi che effettuano, e queste sono, metodi che utilizzano caratteristiche o artefatti basati sui pixel e sul bitstream, e metodi che usano statistiche delle trasformazione dei coefficienti.

#### 2.3.3.1 Caratteristiche spaziali e parametri di codifica

Come detto, la qualità video relativa ad alcune caratteristiche può essere calcolata tramite la misura di artefatti provenienti sia dai pixel che dal bitstream, mettendo insieme i risultati si riesce a calcolare la qualità complessiva. Viene presentato in [36] un metodo che si concentra sulla quantificazione della percezione della qualità di video con codifica H.264/AVC, degradandolo con la perdita dei pacchetti in reti IP. L'errore introdotto a causa della perdita di pacchetti si propaga a causa delle due tipologie di predizione che sono coinvolte nella codifica H.264/AVC, vale a dire, intra-prediction (spaziale) e inter-prediction (temporali). Altri errori possono essere introdotti mentre il decoder cerca di ricostruire al meglio i residui di previsione e/o il motion vectors persi a causa di pacchetti mancanti nel bitstream IP. Per simulare le condizioni di perdita di pacchetti, viene impostata una percentuale di pacchetti persi che varia nell'intervallo [0.1, 20]% con pattern di errore generati utilizzando un modello di Gilbert a due stati ed è stata utilizzata una lunghezza di tre pacchetti. Quantitativamente, le misure effettuate nel modello del metodo proposto tengono conto degli errori dovuti ad occultamenti, errori di propagazione a causa della perdita di MB di riferimento, e anche il degrado introdotto dal canale a causa di specifiche tecniche di codifica del H.264/AVC. I calcoli di queste distorsioni vengono fatte a livello di macroblocchi, e i risultanti valori vengono sommati al frame e ai livelli di sequenza. E' stato osservato che il metodo proposto fornisce risultati che portano ad una buona correlazione con i risultati del SSIM [37]

#### 2.3.3.2 Statistiche delle trasformazione dei coefficienti

In alcuni casi, la trasformazione dei coefficienti può essere ottenuta attraverso la decodifica parziale dei dati codificati del bitstream e da alcune features del bitstream nel dominio spaziale che possono essere combinate per la stima della qualità. Un esempio di questa tecnica si trova in [38], nel quale si cerca di stimare il PSNR per video codificati in MPEG-2 utilizzando i coefficienti DCT. Una prima versione di questo metodo era stata fatta cercando di modellizzare i coefficienti DCT come una funzione di densità di probabilità laplaciana per calcolare il PSNR di ogni tipo di frame uno ad uno, vale a dire i frame I, P, B. Tuttavia, mancava in accuratezza nella valutazione per i frame B. Pertanto, gli autori spiegavano che ciò si causava a causa di una caduta di quantità d'informazioni dei coefficienti DCT presenti nei frame B a causa di un controllo di frequenza e compesazione della motion. Un approccio ibrido per risolvere questo problema è stato trovato in [38] che in aggiunta ai coefficienti DCT è stata utilizzata l'energia dell'immagine. Ciò

ha portato ad un significativo miglioramento con la correlazione con la stima e l'effettivo PSNR, in questi casi il metodo proposto era stato testato su sequenze di SDTV e HDTV.

## 2.4 Machine Learning

Alcuni dei metodi visti in precedenza utilizzano tecniche di Machine learning per il riconoscimento di artefatti, per il calcolo di parametri ausiliari o anche per il calcolo diretto della qualità inserendo come input alcune importanti features. Utilizzando queste potenti tecniche, c'è stato un importante miglioramento dei risultati in correlazione con i risultati delle metriche FR come SSIM. Anche in questo elaborato è stata utilizzata una tecnica di machine learning, per rendere l'algoritmo dinamico e adattarsi ad ogni tipo di video, grazie ad uno studio delle caratteristiche dei contenuti del video. In questa sezione daremo una breve introduzione delle principali tecniche di machine learning ed il loro utilizzo. Per ulteriori dettagli si rimanda al libro *Foundations of Machine Learning*[48]
La machine learning rappresentano una delle aree fondamentali dell'intelligenza artificiale e si occupa della realizzazione di sistemi e algoritmi che si basa su osservazioni come dati per la sintesi di nuova conoscenza. L'apprendimento può avvenire catturando caratteristiche di interesse provenienti da esempi, strutture dati o sensori, per analizzarle e valutarne le relazioni tra le variabili osservate. Uno degli obiettivi principali di ricerca sull'apprendimento automatico è quello di imparare a riconoscere automaticamente modelli complessi e prendere decisioni intelligenti basate su dati; la difficoltà sta nel fatto che l'insieme di tutti i possibili comportamenti dati tutti gli input possibili è troppo grande per essere coperto da insiemi di esempi osservati (dati di allenamento). Da qui è necessario l'utilizzo di tecniche per generalizzare gli esempi citati, in modo da essere in grado di produrre un comportamento utile per casi nuovi.
Al giorno d'oggi non si è ancora in grado di riprodurre sistemi di apprendimento automatico comparabile a quello umano. Tuttavia sono stati inventati algoritmi efficaci per alcuni tipi di compiti di apprendimento, così significative applicazioni commerciali hanno iniziato a comparire. Nel campo conosciuto come data mining, questi algoritmi sono utilizzati di routine per scoprire preziose conoscenze da grandi basi di dati commerciali contenenti un grande numero di informazioni.
Esistono sostanzialmente tre tipi di tecniche di apprendimento: l'apprendimento supervisionato; l'apprendimento non supervisionato; l'apprendimento per rinforzo. Nell'apprendimento supervisionato il training data è composto

da una coppia di esempi determinata da un oggetto di input (tipicamente un vettore) e un valore di output desiderato (anche chiamato supervisory signal). Un algoritmo di apprendimento supervisionato genera una funzione di inferenza (classificatore) che dovrebbe essere in grado di predire il corretto valore di output per ogni input valido. Nell'apprendimento non supervisionato il problema diventa quello di trovare strutture nascoste in strutture dati non preclassificate da cui non è possibile valutare una possibile soluzione. L'apprendimento non supervisionato è strettamente collegato al problema di stima di densità in statistica. La tecnica di programmazione dell'apprendimento per rinforzo si basa sul presupposto di potere ricevere degli stimoli dall'esterno a seconda delle scelte dell'algoritmo. Gli algoritmi per il reinforcement learning tentano di determinare una politica tesa a massimizzare gli incentivi cumulati ricevuti dall'agente nel corso della sua esplorazione del problema. L'apprendimento con rinforzo differisce da quello supervisionato poiché non sono mai presentate delle coppie input-output di esempi noti, né si procede alla correzione esplicita di azioni subottimali.

### 2.4.1 Neural Networks

Una delle tecniche di machine learning più utilizzate, è l'uso di una rete neurale artificiale ANN (artificial neural network), che è anche la tecnica che è stata utilizzata in questo elaborato. Le neural network sono modelli matematici che si basano sulle reti neurali biologiche e rappresentano l'interconnessione tra elementi definiti neuroni artificiali, ossia costrutti matematici che in qualche misura imitano le proprietà dei neuroni viventi. Questi modelli matematici possono essere utilizzati sia per ottenere una comprensione delle reti neurali biologiche, ma ancor di più per risolvere problemi ingegneristici di intelligenza artificiale come quelli che si pongono in diversi ambiti tecnologici (in elettronica, informatica, simulazione, e altre discipline). Tale modello è costituito da un gruppo di interconnessioni di informazioni costituite da neuroni artificiali e processi che utilizzano un approccio di connessionismo di calcolo figura (2.8). Nella maggior parte dei casi una rete neurale artificiale è un sistema adattivo che cambia la sua struttura sulla base di informazioni esterne o interne che scorrono attraverso la rete durante la fase di apprendimento. In termini pratici le reti neurali sono strutture non-lineari di dati statistici organizzate come strumenti di modellazione. Esse possono essere utilizzate per simulare relazioni complesse tra ingressi e uscite che altre funzioni analitiche non riescono a rappresentare. Una rete neurale artificiale riceve segnali esterni su uno strato di nodi (unità di elaborazione) d'ingresso, ciascuno dei quali è collegato con numerosi nodi interni, organizzati in più livelli. Ogni nodo elabora i segnali ricevuti e trasmette il risultato a nodi

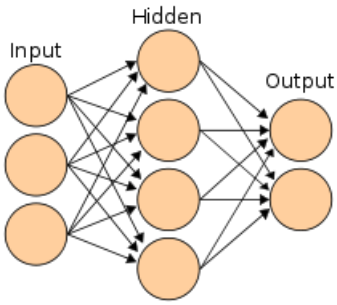


Figura 2.8: Rete Neurale Artificiale

successivi.
Le reti neurali si basano principalmente sulla simulazione di neuroni artificiali opportunamente collegati. I neuroni ricevono in ingresso degli stimoli e li elaborano. L'elaborazione può essere anche molto sofisticata ma in un caso semplice si può pensare che i singoli ingressi vengano moltiplicati per un opportuno valore detto peso, il risultato delle moltiplicazioni viene sommato e se la somma supera una certa soglia il neurone si attiva attivando la sua uscita. Il peso indica l'efficacia sinaptica della linea di ingresso e serve a quantificarne l'importanza. Un ingresso molto importante avrà un peso elevato, mentre un ingresso poco utile all'elaborazione avrà un peso inferiore. Si può pensare che se due neuroni comunicano fra loro utilizzando maggiormente alcune connessioni allora tali connessioni avranno un peso maggiore, fino a che non si creeranno delle connessioni tra l'ingresso e l'uscita della rete che sfruttano "percorsi preferenziali". Tuttavia è sbagliato pensare che la rete finisca col produrre un unico percorso di connessione: tutte le combinazioni infatti avranno un certo peso, e quindi contribuiscono al collegamento ingresso/uscita.

Il modello in figura 2.9 rappresenta una classica rete neurale pienamente connessa. I singoli neuroni vengono collegati alla schiera di neuroni successivi, in modo da formare una rete di neuroni. Normalmente una rete è formata da tre strati: Nel primo abbiamo gli ingressi (I), questo strato si preoccupa di trattare gli ingressi in modo da adeguarli alle richieste dei neuroni. Se i segnali in ingresso sono già trattati può anche non esserci; Il secondo strato è quello nascosto (H, hidden), si preoccupa dell'elaborazione vera e propria e può essere composto anche da più colonne di neuroni; Il terzo strato è quello di uscita (O) e si preoccupa di raccogliere i risultati ed adattarli alle richieste del blocco successivo della rete neurale. Queste reti possono essere anche molto complesse e coinvolgere migliaia di neuroni e decine di migliaia di connessioni.

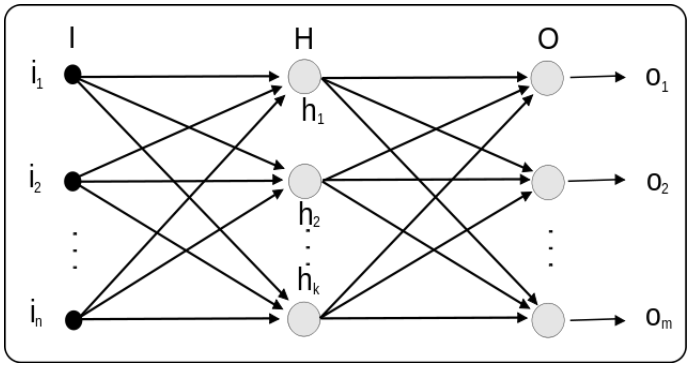


Figura 2.9: Rete Neurale pienamente connessa [2.4.1]

Per costruire la struttura di una rete neurale multistrato si possono inserire strati Hidden; vi sono però alcune dimostrazioni che mostrano che con 1 o 2 strati di tipo Hidden si ottiene una stessa efficace generalizzazione da una rete rispetto a quella con più strati Hidden. L'efficacia di generalizzare di una rete neurale multistrato dipende ovviamente dall'addestramento che ha ricevuto e dal fatto di essere riuscita o meno ad entrare in un minimo locale buono.
Le reti neurali per come sono costruite lavorano in parallelo e sono quindi in grado di trattare molti dati. Si tratta in sostanza di un sofisticato sistema di tipo statistico dotato di una buona immunità al rumore; se alcune unità del sistema dovessero funzionare male, la rete nel suo complesso avrebbe delle riduzioni di prestazioni ma difficilmente andrebbe incontro ad un blocco del sistema. I software di ultima generazione dedicati alle reti neurali richiedono comunque buone conoscenze statistiche; il grado di apparente utilizzabilità immediata non deve trarre in inganno, pur permettendo all'utente di effettuare subito previsioni o classificazioni, seppure con i limiti del caso. Da un punto di vista industriale, risultano efficaci quando si dispone di dati storici che possono essere trattati con gli algoritmi neurali. Ciò è di interesse per la produzione perché permette di estrarre dati e modelli senza effettuare ulteriori prove e sperimentazioni.
I modelli prodotti dalle reti neurali, anche se molto efficienti, non sono spiegabili in linguaggio simbolico umano: i risultati vanno accettati "così come sono", da cui anche la definizione inglese delle reti neurali come "black box": in altre parole, a differenza di un sistema algoritmico, dove si può esaminare passo-passo il percorso che dall'input genera l'output, una rete neurale è in grado di generare un risultato valido, o comunque con una alta probabilità di essere accettabile, ma non è possibile spiegare come e perché tale risultato sia stato generato. Come per qualsiasi algoritmo di modellazione, anche le reti

neurali sono efficienti solo se le variabili predittive sono scelte con cura. Non sono in grado di trattare in modo efficiente variabili di tipo categorico (per esempio, il nome della città) con molti valori diversi. Necessitano di una fase di addestramento del sistema che fissi i pesi dei singoli neuroni e questa fase può richiedere molto tempo, se il numero dei record e delle variabili analizzate è molto grande. Non esistono teoremi o modelli che permettano di definire la rete ottima, quindi la riuscita di una rete dipende molto dall'esperienza del suo creatore.

# Capitolo 3

# Metodo Proposto

In questo capitolo viene descritto in dettaglio il metodo proposto tramite dei diagrammi a blocchi dei sotto-metodi utilizzati per le varie misure. Verrà spiegato in dettaglio ogni singolo modulo che andrà a comporre il software nella sua completezza. Inoltre, verrà mostrato l'intero processo di stima della qualità, come un frame verrà catturato, in che modo il frame verrà analizzato, e di conseguenza la metrica per stimare la qualità complessiva dell'intero video. Sarà spiegato il perché dell'utilizzo delle tecniche di machine learning, in questo caso è stata scelta un rete neurale artificiale, e come queste sono state integrate al sistema per migliorare i risultati.

## 3.1 Stima della qualità frame per frame

In questo elaborato, viene presentato un nuovo metodo No-reference per la stima della qualità video, esaminando il degrado della qualità frame per frame al fine di stimare la qualità complessiva del video. Questo metodo può essere utilizzato per automatizzare la stima del QoE e può inoltre, essere eseguito anche in dispositivi mobile di ultima generazione grazie alla sua semplicità computazionale e al progresso tecnologico. Il metodo, seguendo la classificazione spiegata nella sezione 2.3, rientra nella categoria dei metodi Ibridi che utilizzano le informazioni provenienti dal bitstream e dai pixel per la stima della qualità. Come detto, il metodo basandosi sull'analisi dei frame, grazie a una combinazione lineare di blur e noise, viene misurata la qualità, esso prevede anche l'uso di una **Artificial Neural Network (ANN)** 2.4.1, che guidata dall'informazione contenuta nel bitstream seleziona in maniera appropriata importanti parametri chiamati pesi. Questi pesi vengono associati al blur e al noise e rappresentano quanto blur e noise per un particolare video influenzano la qualità del video. Da notare che questi parametri cambiano

da video a video ed è pertanto necessario trovare un metodo che, prescidendo dalle specificità di video individuali, abbia validità generale. La bassa complessità e la velocità di calcolo dell'approccio presentato lo rende adatto a valutazioni in tempo reale sulla rete. Il metodo è stato sviluppato prendendo spunto dal metodo proposto in [6], focalizzando il problema nel cercare di rendere il metodo più generico possibile e quindi stimare in maniera dinamica i pesi fondamentali anche nell'algoritmo originale. A questo scopo, viene utilizzata una **Artificial Neural Network (ANN)** che prendendo in ingresso alcuni dati contenuti nel bitstream del video, restituisce come output i pesi necessari alla stima della qualità. Il metodo viene testato in real-time, in uno scenario di video streaming end-to-end dove la rete viene emulata e l'obiettivo è quello di approssimare la metrica FR SSIM (sez. 2.2.3) considerata lo stato dell'arte per la stima della qualità. I risultati mostrano di avere una alta correlazione tra il metodo proposto e la metrica FR (utilizzata come benchmark). In questa sezione presenteremo le tecniche utilizzate, la funzionalità del metodo, le assunzioni fatte e i punti di forza.

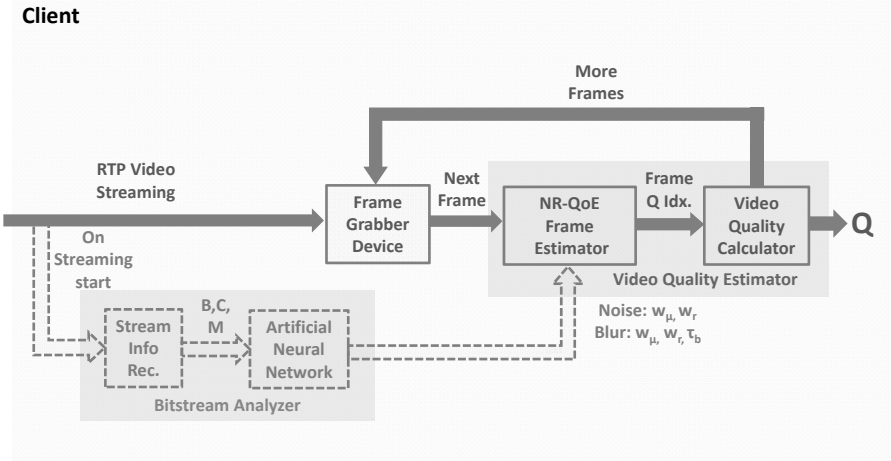


Figura 3.1: Diagramma a blocchi del metodo NR-Video proposto per la stima della qualità

## 3.2 Composizione del metodo

La figura 3.1 raffigura il diagramma a blocchi del metodo proposto. Durante una sessione, un server RTP video streaming comincia a trasmettere il video al client. Una volta ricevute le informazioni contenute nel header del video, queste vengono processate dal **Bitstream Analyzer (BA)** che,

in base ai dati processati, fornirà gli input richiesti dal prossimo blocco il **Video Quality Estimator (VQE)**. Tutti questi dati saranno utilizzati in parallelo durante le fasi di trasmissione e visualizzazione del video. L'elaborazione nella fase di **BA** verrà eseguita soltanto una volta durante l'intera sessione, precisamente quando si comincia la sessione di trasmissione, una volta ottenuti i dati necessari non verrà più richiesta. Dopo aver cominciato la fase di trasmissione dei frames, questi nel momento in cui arriveranno dallo streaming verranno in primo luogo catturati dal **Frame Grabber Device (FGD)** e dopo elaborati. Questo blocco da la possibilità di cambiare il rate dei frame da analizzare per il video stream. Questa caratteristica è stata pensata per garantire la regolare valutazione della qualità anche in contesti on-line, ed inoltre questo rate dovrebbe essere settato anche in riguardo alla potenza di calcolo del dispositivo client, in quanto un rate elevato porterebbe a tempi di calcolo eccessivi per un eventuale dispositivo client con risorse di calcolo limitate, anche se fornirebbe un risultato più accurato in quanto si andrebbe a processare più frame. D'altro canto, un rate basso velocizzerebbe il metodo ma ne risentirebbe in accuratezza dei risultati andando a processare un minor numero di frame. Una volta catturato il frame viene consegnato al **VQE**. Il VQE provvederà alla stima della qualità del frame e riporterà il risultato per la stima della qualità complessiva del video; dopodiché procederà con la richiesta del prossimo frame. Quando lo streaming finirà, il metodo restituirà la qualità video calcolata. Nelle prossime sezione andremo a spiegare nel dettaglio i blocchi appena descritti.

### 3.2.1 Stimatore della qualità video

il *Video Quality Estimator* **(VQE)** può essere suddiviso in due blocchi, *NR-Quality-Frame Estimator* **(NFE)** e il *Video Quality Calculator* **(VQC)**. Il primo è il modulo principale, che stima la qualità del frame basandosi su caratteristiche dei pixel, esattamente, misura la quantità di blur e noise che degradano la qualità. Il secondo include la qualità del frame calcolata nella qualità media del video e richiede il prossimo frame al **FGD**.
Rumore, blur, tonalità e contrasto possono essere introdotte o alterate durente le varie fasi della elaborazione del video o dell'immagine, che possono essere la compressione, la trasmissione, la visualizzazione, e così via. Tuttavia, è usuale assumere che i più importanti fattori per il degrado della qualità sono il blur e il noise. Basandosi su questa premessa, Choi et all. designarono una metrica NR per la stima della qualita delle immagini [6]. Nello sviluppo del metodo proposto in questo elaborato si è seguito il lavoro di Choi (che era limitato solo alle immagini), adattando il loro algoritmo al video streaming. L'equazione 3.1 mostra la metrica che stima la qualità come una combinazio-

ne lineare dei contributi del blur, media e rapporto, e noise, media e rapporto, pesati da differenti valori costanti.

$$Q = 1 - (w_{\mu_b} * \mu_b + w_{r_b} * r_b + w_{\mu_n} * \mu_n + w_{r_n} * r_n) \tag{3.1}$$

L'algoritmo 1 descrive il calcolo delle componenti del blur, necessari all'equazione 3.1 per la stima della qualità. Per prima cosa viene calcolato il valore della differenza tra i pixel e la sua media, effettuando questa operazione con i pixel vicini verticalmente e con i pixel vicini orizzontalmente, ottenendo così due medie quella verticale e quella orizzontale, rispettivamente. Questi valori vengono utilizzati per catturare i pixel che formano gli *edge* nel frame che saranno usati per decidere se un pixel è affetto da blur oppure no. Come già detto, il calcolo viene fatto in entrambe le direzioni dei pixel. Dopodiché, viene stimato il *Inverse Blurriness Index*, derivato dal massimo tra il blur calcolato in direzione verticale e orizzontale per uno stesso pixel, che poi verrà confrontato con la soglia del blur $\tau_b$ per decidere se il pixel in esame è affetto da blur oppure no. Anche questa soglia $\tau_b$ è dinamica e varia da video a video, come già detto verrà decisa dalla **ANN** insieme ai pesi. In fine, viene calcolato $\mu_b$ come la somma dei Inverse Blurriness Index diviso il numero dei pixel del frame affetto da blur e il $r_n$ come il rapporto tra il numero di pixel affetti da blur e il numero di pixel presenti nei edge.

Riguardo la stima del rumore (algoritmo 2), siccome il ritrovamento dei *edge* può essere influenzata dalla presenza di rumore, i frames vengono preprocessati attraverso un filtro per il rumore prima della fase di ritrovamento dei *edge*. Ciò viene fatto utilizzando un filtro mediano. Successivamente, i pixel che formeranno gli *edge* verranno rilevati nel frame filtrato. Il rilevamento viene effettuato seguendo la stessa procedura della fase di stima del blur. Una volta ottenuti gli *edge* in entrambe le direzione verticali e orizzontali, si procede al calcolo dei pixel corrotti dal rumore. Il calcolo di $\mu_n$ viene eseguito come la somma dei valori dei pixel con rumore diviso il numero di pixel affetti da rumore, mentre $r_n$ viene calcolato come il numero di pixel corrotti dal rumore diviso il numero totale dei pixel presenti nel frame.

La figura 3.2 mostra come è stato implementato il **NFE**, la metrica spiegata precedentemente. Prima di tutto, il frame viene diviso in quattro quadranti, che vengono analizzati simultaneamente in termini di blur e noise usando le procedure mostrate nell'algoritmo 1 e 2. I risultati verranno utilizzati nella metrica per la stima della qualità in accordo con l'equazione 3.1, si otterranno così quattro stime di qualità, ognuna per ogni quadrante. Infine, la qualità del frame è ottenuta dalla media dei risultati dei quattro quadranti, questa procedura migliora la velocità del metodo e lo rende applicabile in un contesto reale di valutazione online. I pesi $w_{\mu_b}$, $w_{r_b}$, $w_{\mu_n}$, $w_{r_b}$ e la soglia del

**Algorithm 1** Blur ($\mu_b$, $r_b$)

**1. Calcolo della differenza dei valori dei pixel e le medie**

$\forall(i,j) \in (M,N)$

$$\begin{aligned} D_h(i,j) &= |f(i,j+1) - f(i,j-1)| \\ D_v(i,j) &= |f(i+1,j) - f(i-1,j)| \end{aligned} \tag{3.2}$$

$$\begin{aligned} D_{h-mean} &= \frac{1}{M*N} * \sum_{i=1}^{M}\sum_{j=1}^{N} D_h(i,j) \\ D_{v-mean} &= \frac{1}{M*N} * \sum_{i=1}^{M}\sum_{j=1}^{N} D_v(i,j) \end{aligned} \tag{3.3}$$

**2. Ottenimento dei pixel candidati a formare un *edge***

$\forall(i,j) \in (M,N)$

$$\begin{aligned} C_h(i,j) &= \begin{cases} D_h(i,j), & \text{if } D_h(i,j) > D_{h-mean} \\ 0, & \text{otherwise} \end{cases} \\ C_v(i,j) &= \begin{cases} D_v(i,j), & \text{if } D_v(i,j) > D_{v-mean} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{3.4}$$

**3. Ottenimento dei pixel che formano un *edge* ($E_h(i,j)$,$E_v(i,j)$)**

$\forall(i,j) \in (M,N)$

$$\begin{aligned} E_h(i,j) &= \begin{cases} 1, & \text{if } C_h(i,j) > C_h(i,j+1) \\ & \text{and } C_h(i,j) > C_h(i,j-1) \\ 0, & \text{otherwise} \end{cases} \\ E_v(i,j) &= \begin{cases} 1, & \text{if } C_v(i,j) > C_v(i,j+1) \\ & \text{and } C_v(i,j) > C_h(i,j-1) \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{3.5}$$

**4. Analisi dei pixel con blur**

$\forall(i,j) \in (M,N)$

$$\begin{aligned} BR_h(i,j) &= \frac{|f(i,j) - 0.5 * D_h(i,j)|}{0.5 * D_h(i,j)} \\ BR_v(i,j) &= \frac{|f(i,j) - 0.5 * D_v(i,j)|}{0.5 * D_v(i,j)} \end{aligned} \tag{3.6}$$

**5. Ottenimento del Inverse Blurriness Index**

$\forall(i,j) \in (M,N)$

$$\begin{aligned} \mathcal{I}_b(i,j) &= \max(BR_h(i,j), BR_v(i,j)) \\ B(i,j) &= \begin{cases} 1, & \text{if } \mathcal{I}_b(i,j) < \tau_b \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{3.7}$$

**7. Calcolo di $\mu_b$ e $r_b$**

$$\mu_n = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} \mathcal{I}_b(i,j)}{\sum_{i=1}^{M}\sum_{j=1}^{N} B(i,j)} \qquad r_b = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} B(i,j)}{\sum_{i=1}^{M}\sum_{j=1}^{N} (E(i,j))} \tag{3.8}$$

**Algorithm 2** Noise ($\mu_n$, $r_n$)

**1. Applicazione del filtro medio al frame**

$\forall(i,j) \in (M,N)$

$$g(i,j) = \frac{1}{3x3}[\sum_{t=-1}^{1}\sum_{z=-1}^{1} f(i+t, j+z)] \tag{3.9}$$

**2. Calcolo della differenza dei valori dei pixel e le medie**

$\forall(i,j) \in (M,N)$

$$\begin{aligned} D_h(i,j) &= |g(i,j+1) - g(i,j-1)| \\ D_v(i,j) &= |g(i+1,j) - g(i-1,j)| \end{aligned} \tag{3.10}$$

**3. Ottenimento dei pixel candidati ad avere rumore**

$\forall(i,j) \in (M,N)$

$$N_c(i,j) = \begin{cases} \max(D_h(i,j), D_v(i,j)), & \text{if } D_h(i,j) \leq D_{h-mean} \\ & \text{and } D_v(i,j) \leq D_{v-mean} \\ 0, & \text{otherwise} \end{cases} \tag{3.11}$$

**4. Decisore se il pixel è affetto da rumore**

$$N_{c-mean} = \frac{1}{M*N}\sum_{i=1}^{M}\sum_{j=1}^{N} N_{cand}(i,j) \tag{3.12}$$

$\forall(i,j) \in (M,N)$

$$N(i,j) = \begin{cases} N_c(i,j), & \text{if } (N_c(i,j) > N_{c-mean}) \\ 0, & \text{otherwise} \end{cases} \tag{3.13}$$

**5. Calcolo del $\mu_n$ e $r_n$**

$$\mu_n = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} N(i,j)}{Count_n} \qquad r_b = \frac{Count_n}{M*N} \tag{3.14}$$

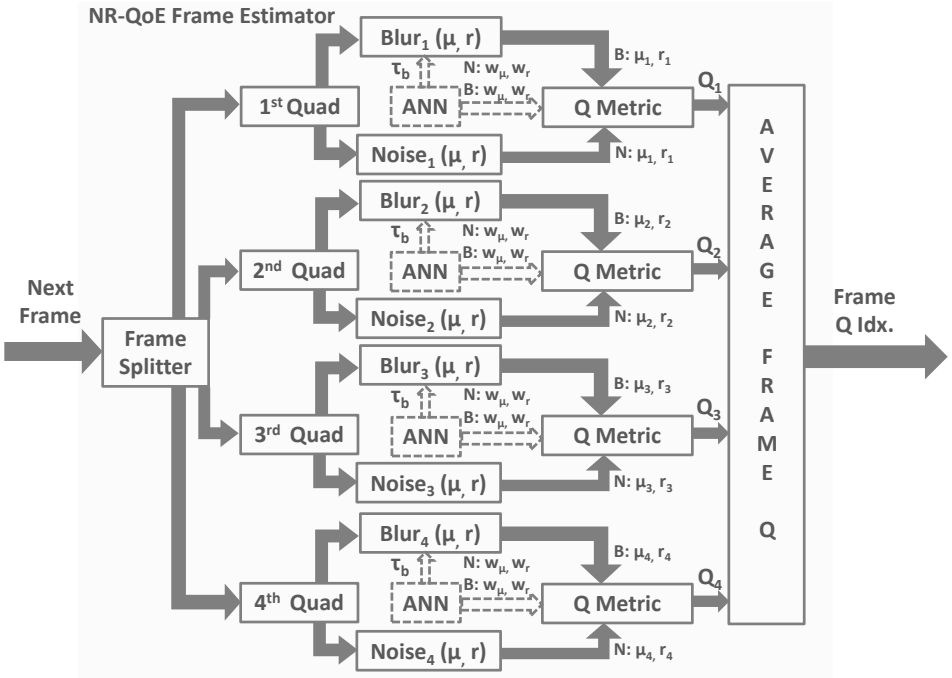


Figura 3.2: Stimatore frame NR-QoE

blur ($Th_b$) verranno fornite dalla rete neurale artificiale (**ANN**) contenuta nel **BA** alla metrica per la stima della qualità e all'algoritmo per la stima del blur rispettivamente, che presenteremo nella prossima sottosezione.

### 3.2.2 Analizzatore del Bitstream

Al fine di stimare la qualità video in maniera più accurata possibile, le costanti dei pesi $w_{\mu_b}$, $w_{r_b}$, $w_{\mu_n}$, $w_{r_b}$ e la soglia del blur ($Th_b$) devono essere adattati al tipo di video in streaming. La determinazione di costanti imprecise possono causare una sbagliata stima della qualità, e nella peggiore delle ipotesi dare giudizi sbagliati sulle performance della rete. Gli autori di [6] applicano una regressione lineare per stimare le costanti univoche nel caso delle sole immagini. Tuttavia, i video contengono parametri codificati che hanno una significativa influenza non-lineare sulla qualità, e richiedono tecniche più complesse. Lo scopo del BA è quello, di valutare con precisione i pesi e la soglia per essere usati nel **VQE**, dati i parametri codificati del video stream. Il BA è diviso in due blocchi (Figura:3.1) lo *Stream Information Recorder* (**SIR**) e la *Artificial Neural Network* (**ANN**).

#### 3.2.2.1 Stream Information Recorder (SIR)

Questo modulo serve per ottenere le informazioni necessarie alla ANN dal header del bitstream codificato. Per descrivere per bene questo modulo c'è bisogno di spiegare in dettaglio i dati utilizzati in input nella ANN. Un video stream può essere caratterizzato da diversi parametri e, in base al tipo di video, questi parametri possono variare considerevolmente. La qualità può essere dipendente dal numero di bit trasmessi per un intervallo di tempo, ad esempio la bitrate. Incrementando la bitrate si può avere un diretto incremento della qualità percepita. Inoltre, parametri che riguardano la composizione della scena del video, è stato dimostrato che hanno una grande influenza su larga scala della qualità percepita [39]. La scena può essere caratterizzata dalla *Complexity* e dal livello di *Motion*, definiti come il numero di oggetti o elementi presenti nella scena e la quantità di movimento presente nel video, rispettivamente [40]. Quindi, alla luce delle ricerche effettuate, sono state scelti i seguenti parametri: Bitrate ($B$), livello di Motion ($M$) e Complexity della scena ($C$), come input alla ANN, per la rappresentazione delle caratteristiche del video. Il modulo SIR analizza l'header del bitstream in arrivo per calcolare questi parametri. ($B$) può essere direttamente recuperato dai codec del video, mentre ($C$) e ($M$) vengono calcolati in accordo con le equazioni 3.15 [39].

$$C = \frac{Bits_I}{2 * 10^6 * 0.91^{QP_I}} \qquad M = \frac{Bits_P}{2 * 10^6 * 0.87^{QP_P}} \tag{3.15}$$

dove $Bits_I$, $Bits_P$ sono i bits relativi ai frame codificati *Intra (I)* e *Inter (P)*, $QP_I$, $QP_P$ rappresentano la media dei parametri quantizzati dei *I-Frame* e *P-Frame*, rispettivamente. Tutti questi valori sono ottenibili direttamente dal processo di codifica e quindi, non si ha un incremento della complessità computazionale o del tempo d'esecuzione del metodo.

#### 3.2.2.2 Artificial Neural Network (ANN)

Le stime delle costanti principali, come i pesi e la soglia del blur spiegati in precedenza, vengono ottenuti utilizzando una **artificial neural network (ANN)**, con input $B$, $M$ e $C$ fornita dall'unità SIR. La ragione per aver scelto di utilizzare una ANN contro un'altra tecnica di machine learning è dovuta alla semplicità di mappare i dati in input in output e alla grande resistenza al rumore cioè a dati incongruenti presenti nel data-set. Inoltre, possono anche lavorare con dati in input incompleti o con alcuni valori fuori norma, dopo essere state allenate soltanto con un data-set molto ridotto.
Come spiegato nella sezione 2.4.1, una ANN consiste in un gruppo di processi base di unità interconnesse tra loro, chiamate più comunemente *Neuroni*.

Ogni coppia di neuroni è connessa tra loro, a questa connessione viene associata un peso generato durante la fase di training. Esistono diversi tipi di possibili reti neurali, il tipo di ANN utilizzata in questo elaborato consiste in una **Multi-Layer Perceptron (MLP)** [41]. Consiste in multipli livelli di nodi in un grafo diretto, ogni livello è pienamente connesso con i livelli successivi. È stata scelta una MLP perché grazie alla sua topologia a strati feed-forward permette alla rete di avere una semplice interpretazione della forma del modello input-output. Inoltre, questo tipo di rete è abbastanza flessibile nella modellizzazione della funzione con una complessità quasi arbitraria [42].
Nella fase di creazione della ANN diversi parametri devono essere definiti con attenzione. Prima di tutto, la selezione del training set è fondamentale. Questo insieme consiste nelle misure dei segnali di input da far corrispondere all'obiettivo, esempio, l'output desiderato. L'adeguatezza del training set alla rete determinerà le performance dell'algoritmo, pertanto, se non dovesse venir scelta con attenzione causerà risultati inaccurati.
Una volta deciso il training set, si procede a definire il numero di neuroni che comporranno la rete. I neuroni sono distribuiti in tre livelli: Input, Hidden, e Output. Definire il numero di neuroni del livello hidden è un processo molto delicato, la linea guida è quella di realizzare la ANN più piccola possibile che rispetti il limite d'errore richiesto [42], esso dipende anche dallo scopo della ANN, dal numero di input e output e anche dal training set. Una rete con troppi neuroni può causare l'effetto di overfitting. Al contrario, scegliere un basso numero di neuroni può causare l'effetto contrario cioè unfit del modello. Inoltre, è possibile aggiungere dei neuroni speciali detti bias, che non sono connessi con il livello precedente e che provvedono ad un output costante, utile per traslare la funzione di attivazione.
Una volta costruita la ANN e scelto il training set, si procede con la selezione del Learning Rate. Il Learning Rate è un valore compreso tra 0 e 1, che governa la velocità di apprendimento della rete dal training set. Un valore basso di Learning Rate richiede un grande numero di cicli, rendendo il processo di apprendimento estremamente lento. Ma, un Learning Rate troppo vicino a 1 può creare delle divergenze dei pesi delle connessioni e l'errore oggettivo subisce delle pesanti oscillazioni.
L'ultimo parametro da prendere in considerazione è il Momentum, usato per incrementare la velocità di apprendimento del sistema. Come per il Learning Rate, il Momentum è compreso tra 0 e 1. Un alto valore incrementa la velocità di convergenza. Tuttavia, se il Momentum è troppo altro il sistema tenderebbe a cadere in un minimo locale, provocando instabilità. Discuteremo nel prossimo capitolo come questi parametri sono stati scelti e la struttura della rete che è stata implementata.

# Capitolo 4

# Implementazione

In questo capitolo verranno descritti la parte implementativa del software e gli strumenti utilizzati per la sua costruzione. Verranno spiegate le librerie utilizzate, il tipo di linguaggio adoperato per la realizzazione del metodo e come sono state implementate tutte le fasi descritte nel capitolo 3. Inoltre, verrà illustrata la fase di avvio della sessione di video streaming, l'ottenimento delle features dal bitstream, la cattura dei frame e la misura delle distorsioni, per arrivare alla vera e propria misura della qualità complessiva.

## 4.1 Librerie utilizzate

Il metodo presentato è stato implementato interamente in Java, in modo da poter realizzare un software che si potesse eseguire anche in dispositivi mobile Android. Anche se Java è definito un linguaggio semi-interpretato[1] e ciò potrebbe aumentare i tempi d'esecuzione, grazie all'avanzamento tecnologico e allo sviluppo di nuovi dispositivi sempre più potenti tutto ciò non è stato più un problema. Anzi, grazie all'utilizzo di Java, e di alcune proprietà che il linguaggio offre, esso ha facilitato la costruzione del software con buoni tempi d'esecuzione. Nella realizzazione del metodo sono stati adoperati diversi strumenti, tra cui le librerie JavaCV[43] e il framework Neuroph Studio[42]. JavaCV fornisce validi strumenti e ottimi algoritmi per la computer vision, mentre Neuroph è un framework che facilita la costruzione e l'utilizzo di vari tipi di reti neurali artificiali (ANN). La realizzazione del software è stata facilitata principipalmente dall'utilizzo degli strumenti appena citati, che verranno presentati in dettaglio nelle prossime due sottosezioni.

[1] Il codice una volta compilato deve essere interpretato dalla Java Virtual Machine

### 4.1.1 JavaCV

Le JavaCV[43] sono delle librerie che mettono a disposizione tecniche di Image/video processing, come ad esempio l'applicazioni di vari tipi di filtri, conversione di colori, analisi dei pixel e così via. Esse sono composte da più librerie (OpenCV, FFmpeg, libdc1394, PGR FlyCapture, OpenKinect, videoInput, ARToolKitPlus, e flandmark), queste sono le librerie più utilizzate dai ricercatori nel campo della computer vision. Le librerie elencate che compongono le JavaCV, sono delle librerie sviluppate per essere utilizzate nei linguaggi di programmazione C++ o Python. Gli autori, per realizzare una versione Java, utilizzano dei wrapper contenuti in un'altra libreria chiamata JavaCPP[44], che in poche parole fornisce interfacce con il codice nativo C++ contenuto in Java, e grazie al loro utilizzo è stato possibile creare una versione di tutte queste librerie in Java. Le JavaCV, rendono disponibili i metodi contenuti nelle classi e facilmente utilizzabili nella piattaforma Java e quindi anche disponibili per sistemi Android.
Le JavaCV forniscono anche acceleratori grafici per la visualizzazione delle immagini (CanvasFrame e GLCanvasFrame), metodi semplici da utilizzare per eseguire codice in parallelo su hardware multi-core (Parallel), metodi per le calibrazioni di colori o geometriche per camere o proiettori (GeometricCalibrator, ProCamGeometricCalibrator, ProCamColorCalibrator), detection e matching di alcune caratteristiche (ObjectFinder), un set di classi che implementano direttamente l'allineamento delle immagini per la proiezione in un sistema di camere (GNImageAligner, ProjectiveTransformer, ProjectiveColorTransformer, ProCamTransformer, e ReflectanceInitializer), un pacchetto per analisi del blob (Blobs). Come è facile intuire le JavaCV offrono una vasta gamma di funzionalità per la computer vision, alcuni di questi metodi provengono anche da altre librerie come OpenCL e OpenGL.
In questo elaborato le librerie comprese in JavaCV che sono state utilizzate maggiormente sono le FFmpeg per la decodifica del video, l'estrapolazione delle informazioni dal bitstream e la cattura dei frame, e le OpenCV per l'analisi dei frame, la stima delle distorsioni e la stima della qualità complessiva.
Le FFmpeg sono librerie per la gestione di dati multimediali, contengono codec per audio e video per le fasi di codifica e decodifica dei video, permette lo streaming dei dati con l'utilizzo di diversi protocolli e fornisce anche degli strumenti utilizzabili da linea di comando per la gestione delle informazioni contenuti nei video.
Le OpenCV sono librerie per programmazione principalmente utilizzate per la computer vision in real-time. Sono librerie sviluppate dall'Intel[2]. Ven-

[2]L'Intel Corporation è la più grande azienda multinazionale produttrice di dispositivi a semiconduttore (microprocessori, dispositivi di memoria, circuiti di supporto alle teleco-

gono utilizzati principalmente con il linguaggio C++ e la loro applicazione include: kit di strumenti per il rilevamento di caratteristiche in 2D e 3D, sistemi di riconoscimento facciale, riconoscimento di gesti, interazioni tra uomo-computer, robotica mobile, identificazione di oggetti, visione stereoscopica, e tante altre applicazioni. In alcune aree, alle OpenCV vengono integrate alcune tecniche di Machine Learning come le Support Vector Machine (SVM) o le Artificial Neural Networks.

### 4.1.2 Neuroph studio

Neuroph[42], è un framework per reti neurali scritto in Java. Serve per sviluppare architetture comuni di reti neurali artificiali. Neuroph semplifica lo sviluppo di reti neurali, fornendo librerie Java neural network e un comodo strumento GUI[3] (chiamato easyNeural) per la creazione, la formazione, la gestione e il salvataggio di reti neurali. Grazie ad esso è possibile creare reti

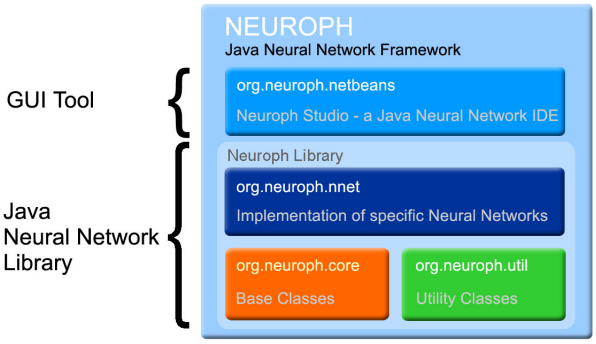


Figura 4.1: Diagramma a blocchi della struttura del Framework Neuroph

neurali personalizzate e la fase di training diventa estremamente semplice. Neuroph si appoggia all'IDE Netbeans per fornire una GUI semplice ed intuitiva 4.2. Le classi principali di Neuroph corrispondono ai concetti basi delle reti neurali come neurone artificiale, livello di neuroni, connessioni neuronali, pesi associati alle connessioni, funzione di trasferimento, funzione di ingresso, regole di apprendimento ecc. Neuroph supporta le architetture più comuni di reti neurali, come la Multilayer Perceptron con Backpropagation,

municazioni e alle applicazioni informatiche) con sede a Santa Clara, California. Fondata nel 1968.

[3]L'interfaccia grafica utente, nota come GUI (dall'inglese Graphical User Interface), comunemente abbreviata in interfaccia grafica, è un tipo di interfaccia utente che consente all'utente di interagire con la macchina controllando oggetti grafici convenzionali.

reti di Kohonen e Hopfield. Tutte queste classi possono essere estese e personalizzate per creare reti neurali personalizzati e le regole di apprendimento. Il framework comprende anche un supporto di riconoscimento di immagini molto sviluppato e operativo.

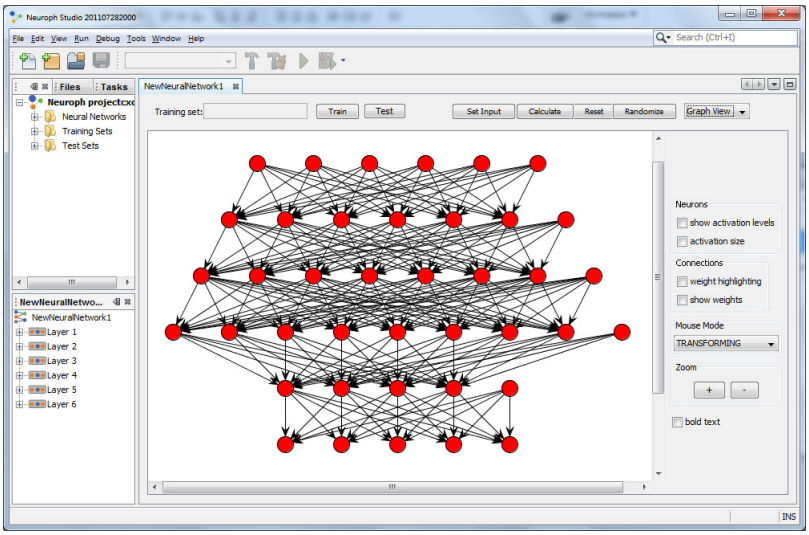


Figura 4.2: GUI di Neuroph con una ANN multilivello pronta per la fase di training

## 4.2 Implementazione Rete Neurale Artificiale

Come già detto, la fase di realizzazione della ANN è stata molto facilitata dall'utilizzo della GUI offerta dal framework Neuroph. Lo scopo dell'utilizzo di una ANN in questo contesto, è dovuto alla ricerca dei parametri ottimali per quanto riguarda la soglia del blur ($Th_b$) e i pesi ($w_{\mu_b}$, $w_{r_b}$, $w_{\mu_n}$, $w_{r_b}$), in base ai contenuti del video in esecuzione. In poche parole rendere l'algoritmo dinamico grazie alla ANN. I parametri scelti per descrivere i contenuti del video in un formato numerico sono: Complexity, Motion e Bitrate, (sottosezione 3.2.2.1). Il framework Neuroph lavora con valori compresi tra 0 e 1. Per quanto riguarda Complexity e Motion non si pone il problema in quanto questi sono valori che, per loro natura, sono compresi tra 0 e 1. Invece per la Bitrate si è dovuto applicare un processo di normalizzazione dei dati. Si è usato un sistema lineare per normalizzare i dati, e viene richiesto il valore minimo e il valore massimo dei dati che si vogliono normalizzare. La formula

utilizzata per la trasformazione dei singoli dati è la seguente 4.1:

$$I = I_{min} + (I_{max} + I_{min}) * \frac{D - D_{min}}{D_{max} - D_{min}} \tag{4.1}$$

dove $I_{min}$, $I_{max}$, sono i valori minimi e massimi che si vogliono ottenere, nel nostro caso $I_{min}$ è 0 mentre $I_{max}$ è 1, rispettivamente. $D_{min}$ e $D_{max}$ sono i valori massimi e minimi dei possibili valori in input. Il software è stato testato con video di bitrate che varia tra 64 kbps e 2048 kbps, quindi $D_{min}$ è 64 mentre $D_{max}$ è 2048. $D$ rappresenta il valore da convertire. Dopo questa operazione si ottengono tutti i valori in input compresi tra 0 e 1, indonei ad essere usati nel framework.
Una volta definiti i dati in input, si procedere alla realizzazione della struttura della ANN. Una comune struttura della ANN è composta dal livello che contiene i neuroni di input, il livello dei neuroni nascosti, che possono essere suddivisi in più sotto-livelli, e il livello dei neuroni di output. Nel nostro caso, il numero di neuroni in input sarà 3 che rappresentano Complexity, Motion e Bitrate, mentre il numero di neuroni in output sono 5, che identificano $Th_b$, $w_{\mu_b}$, $w_{r_b}$, $w_{\mu_n}$, $w_{r_b}$. Discorso diverso per quanto riguarda il livello di neuroni nascosti, come già illustrato nella sezione 2.4.1, è difficile definire a priori il numero di neuroni nascosti. Ciò che si è cercato di fare è stato di cercare la struttura più semplice possibile che mantenesse il tasso di errore più basso possibile e che non appesantisse molto la fase di training. Dopo diversi test si è giunti alla miglior rete con 15 neuroni nascosti più un neurone Bias, disposti su un solo livello. Il tipo di rete scelta fra le varie opzioni disponibili (Fig. 4.3) è una *Multi Layer Perceptron* che, grazie alla sua semplice interpretazione della forma del modello input-output, restituisce ottimi risultati.
Resta da decidere il tipo di funzione di trasferimento da utilizzare e il Learning Rule. Come funzione di trasferimento è stata scelta la *funzione Sigmoid*, in quanto i valori che sono stati utilizzati sono tutti compresi tra 0 e 1. Come Learning Rule l'algoritmo *Backpropagation con Momentum*, che in generale ha un rate di convergenza maggiore di un algoritmo Backpropagation.
Fissati tutti questi parametri (Fig. 4.4) la rete neurale è costruita (Fig. 4.5)ed è pronta per la fase di training. Nelle prossime sottosezioni illustreremo come è stato costruito il Training Set, e le fasi di training e di test della ANN.

### 4.2.1 Costruzione del Dataset

Per essere utilizzata la ANN deve essere allenata, cioè deve essere sottoposta alla fase di training. Quindi, è necessario una raccolta di dati, chiamato Training Set, contenente input e output desiderati, così da permettere alla

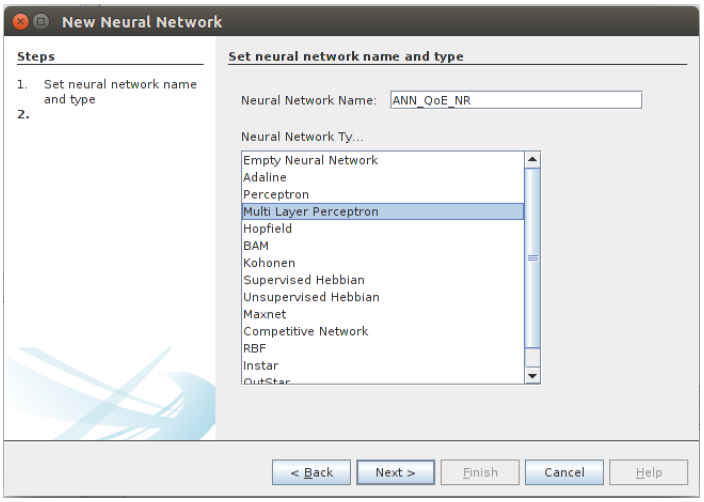


Figura 4.3: Neuroph: selezione del tipo di ANN

ANN di creare un modello sconosciuto, e di trovare una relazione non-lineare tra input e output forniti per avere una funzione predittiva che riesca a fornire output sconosciuti in corrispondenza di quegli input non contenuti nel Training Set.
Il training set utilizzato per allenare la rete è stato creato interamente da zero. Sono stati presi i video contenuti nel database LIVE[45]. Questo database è composto da 10 video differenti, dove ogni video viene proposto a 10 bitrate differenti, ognuno di durata 10 secondi, uno stream a 24 fps e codifica in MPEG-4 AVC/H.264. Gli autori del database LIVE, conducendo vari test soggettivi su ciascuno dei 100 video presenti, mettono a disposizione la media e la deviazione standard del DMOS. I test effettuati dagli autori del database LIVE sono stati eseguiti con differenti tipi di distorsioni per ogni video, come per esempio le distorsioni introdotte dalle trasmissioni wireless o dalla compressione H.264 o MPEG-2. In base ai risultati riportati nel database, relativi ai video con compressione H.264, si è cercato di costruire un appropriato training set, esaminando uno ad uno i video e cercando di stimare gli output desiderati attraverso vari test, modificando di volta in volta i valori di $Th_b$ e $w_{\mu_b}$, $w_{r_b}$, $w_{\mu_n}$, $w_{r_b}$, al fine di ottenere i valori della qualità stimata con il software presentato, correlati con i risultati riportati dagli autori del database LIVE.
L'obiettivo era quello di ottenere un training set che coprisse buona parte del piano definito dalla Motion e Complexity dei video utilizzati, per avere un vasto spettro di generalizzazione per allenare la ANN. La figura 4.6 mostra le caratteristiche dei video presenti nel database LIVE (bs1, mc1, pa1,

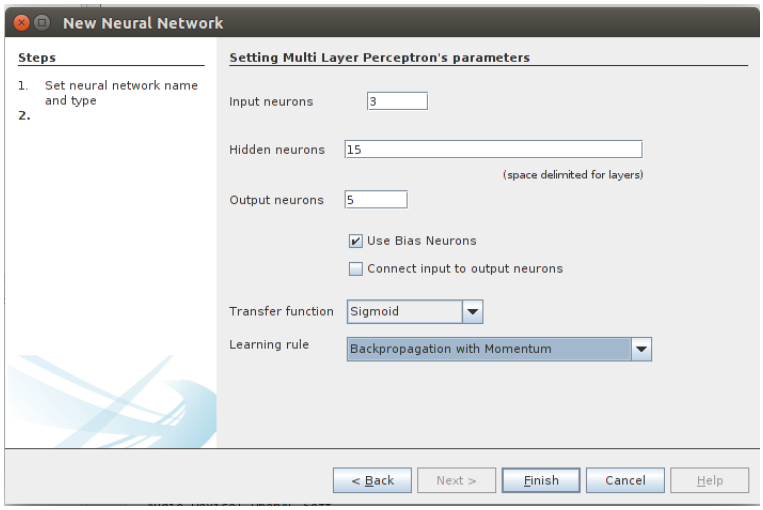


Figura 4.4: Neuroph: selezione dei parametri della struttura ANN

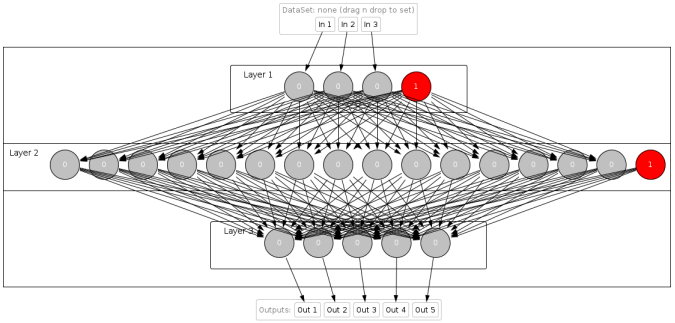


Figura 4.5: Neuroph: rappresentazione logica della struttura ottimale ANN

pr1, rb1, rh1, sf1, sh1, st1, tr1) (Tab. 4.1) e mostra come i diversi valori di Complexity e Motion indicano che i video hanno una diversa dinamica. Per ognuno di questi video sono state riportate Motion e Complexity con le diverse bitrate che variano da 64 Kbps a 2048 Kbps.

Una volta creato il training set, si procede nel caricarlo sul framework Neuroph e a definire alcuni parametri (Fig. 4.7). Un importante parametro è definire il tipo di training set, questo può essere di due tipi, supervisionato o non-supervisionato. Il tipo supervisionato, che è quello più comune, comporta che il training set fornisce i dati in input e per ogni input viene fornito l'output desiderato. La ANN deve costruire la relazione che esiste tra input e output. Per quanto riguarda il metodo non-supervisionato viene fornito solo un pattern dei dati in input. Nel nostro caso utilizziamo il tipo supervisiona-

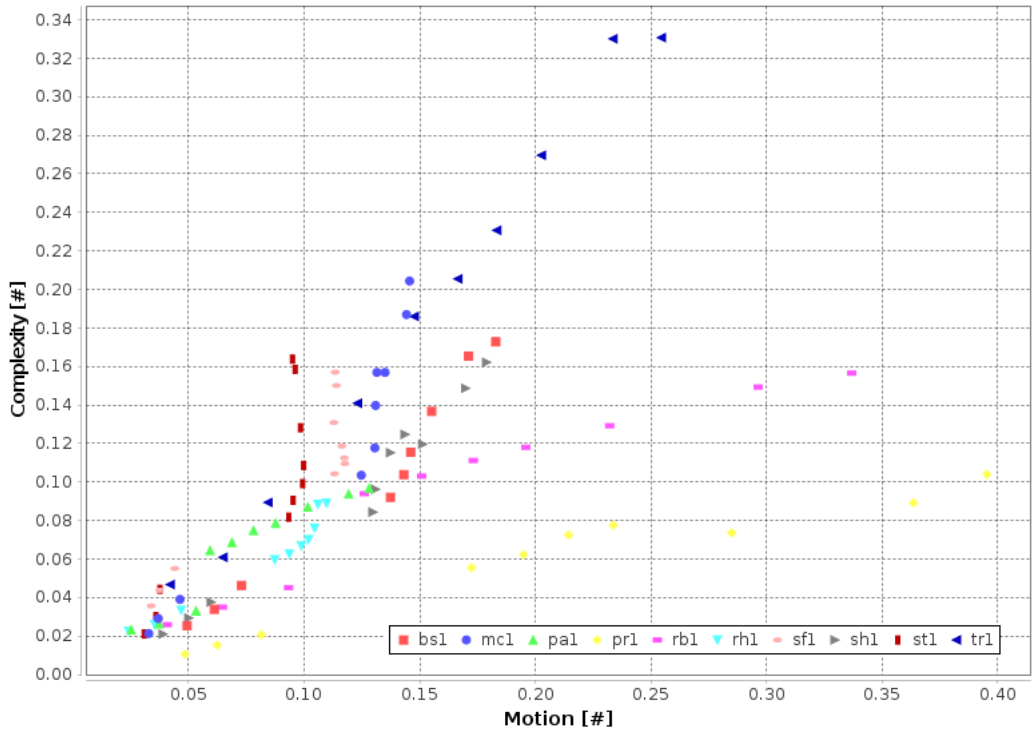


Figura 4.6: Valori di Motion e Complexity del Training Set

to visto che abbiamo tutti i parametri necessari. Dopodiché vengono inseriti il numero di input e il numero di output, ed il framework è pronto per le fasi di training e di testing.

### 4.2.2 Fase di Training e di Testing

Per avviare la procedura di training della ANN bisogna impostare dei parametri importanti. La GUI di Neuroph rende semplice questa operazione, una volta selezionato il training set da utilizzare, si procede alla fase di training. Apparirà una finestra dove è possibile impostare i parametri di learning (Fig. 4.9). In questa finestra è possibile impostare il massimo errore tollerabile, che di default è impostato a 0,01. Durante il processo di training quando l'errore totale della rete sarà inferiore del valore impostato il processo di training sarà completo. Più l'errore è basso più il modello avrà una migliore approssimazione. È possibile impostare il massimo numero di iterazioni e i due parametri di learning fondamentali che governano la velocità di apprendimento della ANN e la sua accuratezza, che sono il learning rate e il

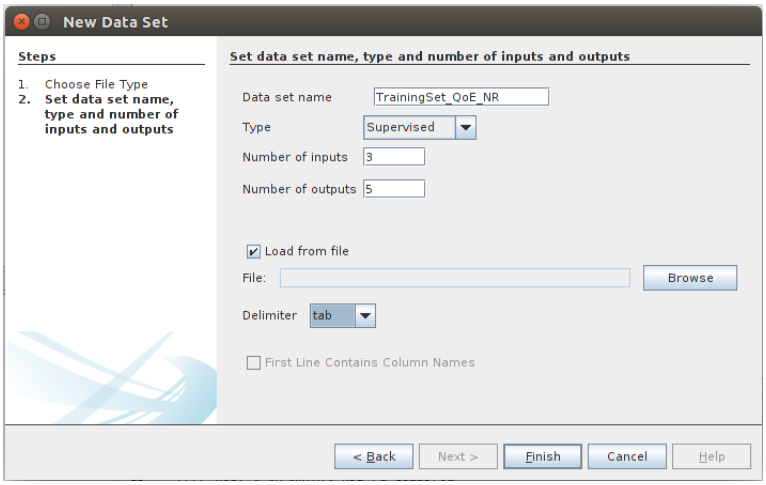


Figura 4.7: Neuroph: selezione dei parametri relativi al Training Set

Training set ... TrainingSet_QoE_NR

| Input 1 | Input 2 | Input 3 | Output 1 | Output 2 | Output 3 | Output 4 | Output 5 |
|---|---|---|---|---|---|---|---|
| 0.0467664144... | 0.0420822745... | 0.0 | 0.3 | 1.0 | 1.0 | 1.0 | 0.5 |
| 0.0609349864... | 0.0649383244... | 0.0322580645... | 0.2 | 1.0 | 1.0 | 0.5 | 0.5 |
| 0.0893982256... | 0.0844051110... | 0.0967741935... | 0.2 | 1.0 | 1.0 | 0.5 | 0.5 |
| 0.1409202191... | 0.1231408395... | 0.1612903225... | 0.2 | 1.0 | 1.0 | 0.5 | 0.5 |
| 0.1860142061... | 0.1476846433... | 0.2258064516... | 0.2 | 1.0 | 1.0 | 0.25 | 0.25 |
| 0.2054936528... | 0.1665518961... | 0.2903225806... | 0.2 | 1.0 | 1.0 | 0.25 | 0.25 |
| 0.2307284957... | 0.1833633784... | 0.3548387096... | 0.1 | 1.0 | 1.0 | 0.2 | 0.2 |
| 0.2696722557... | 0.2026482385... | 0.4838709677... | 0.1 | 0.25 | 0.25 | 0.1 | 0.1 |
| 0.3301536610... | 0.2336325637... | 0.7520161290... | 0.1 | 0.2 | 0.2 | 0.1 | 0.1 |
| 0.3307852873... | 0.2545943002... | 1.0 | 0.1 | 0.1 | 0.2 | 0.2 | 0.15 |
| 0.0213408654... | 0.0328246294... | 0.0 | 0.2 | 1.0 | 1.0 | 1.0 | 0.5 |
| 0.0291452547... | 0.0369268638... | 0.0322580645... | 0.2 | 1.0 | 1.0 | 0.75 | 0.5 |
| 0.0390483991... | 0.0463449083... | 0.0967741935... | 0.2 | 1.0 | 1.0 | 0.5 | 0.4 |
| 0.1035490312... | 0.1247607538... | 0.1612903225... | 0.2 | 1.0 | 1.0 | 0.5 | 0.4 |
| 0.1177461990... | 0.1306247655... | 0.2258064516... | 0.2 | 0.8 | 0.8 | 0.5 | 0.3 |
| 0.1397511351... | 0.1309700414... | 0.2903225806... | 0.1 | 0.8 | 0.8 | 0.5 | 0.3 |
| 0.1569298249... | 0.1315199653... | 0.3548387096... | 0.1 | 0.8 | 0.8 | 0.4 | 0.2 |

Figura 4.8: Neuroph: Training Set completo

momentum. Il learning rate serve a definire ad ogni iterazione quanto variano i pesi associati ai collegamenti tra i neuroni, il momentum è usato per prevenire che un sistema converga ad un minimo locale. Per definire questi valori in maniera ottimale, come anche il numero dei neuroni nascosti, devono essere effettuati diversi test, modificando di volta in volta questi parametri cercando di ottenere il minimo errore e di velocizzare la fase di training.
Un prima configurazione provata prevedeva 8 neuroni nascosti più uno di bias in un unico livello nascosto, il learning rate 0,4 e il momentum impostato a 0,8. Con questa configurazione dopo 50000 iterazioni l'errore non andava mai sotto 0,035. Dopo diversi tentativi, cioè quello di aumentare il numero di neuroni nascosti e di modifiche del learning rate e del momentum, si è arrivati alla configurazione ottimale, costituita da 15 neuroni nascosti più uno di bias, il learning rate fissato a 0,2 e il momentum 0,7. La confi-

| Video | Descrizione |
|---|---|
| tr1 | video dove è presente un trattore che miete il grano |
| sh1 | video dove un uomo indica degli stendardi che scorrono nello sfondo |
| pr1 | persona in bicicletta su uno sfondo di alberi innevati |
| rb1 | presenti delle piccole onde di uno stagno che vanno sulla terra |
| bs1 | inquadratura circolare del cielo con presenza di alberi |
| mc1 | visione di un calendario con allargatura della inquadratura |
| pa1 | percorso pedonale affollato in un centro urbano |
| rh1 | strada urbana intasata |
| sf1 | inquadratura di un ape su un girasole |
| st1 | stazione ferroviaria quasi deserta |

Tabella 4.1: Descrizione dei video presenti nel database LIVE

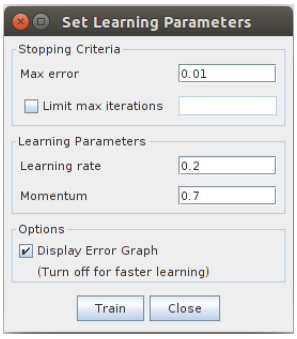


Figura 4.9: Neuroph: Finestra per impostare i parametri di Learning

gurazione appena descritta ha portato ad avere il minimo errore rispetto a tutti gli altri tentativi, cioè, 0,025 dopo 2000 iterazioni. Considerata questa configurazione ottimale si è dato il via alla fase di test. Il framework offre la possibilità di testare la rete allenata con il training set utilizzato, per vedere quanto il modello approssima la realtà d'interesse rappresentato dal nostro training set, e ne viene calcolato l'errore quadratico medio totale che corrisponde a 0.0112. Considerato l'errore accettabile si è utilizzata la suddetta ANN, salvandola nel formato del framework e pronta per essere utilizzata all'interno del software grazie alle API java di Neuroph.

## 4.3 Misura della qualità

In figura 4.10 viene mostrato il diagramma delle classi che rappresenta la struttura del software dal punto di vista implementativo. Si è cercato di

rendere il software più semplice possibile e anche molto scalabile in modo da poter essere applicato in vari dispositivi che supportano la piattaforma virtuale Java, JVM. Il software, come si può vedere dal diagramma, è composto da poche classi ognuna con un compito preciso da realizzare durante la stima della qualità. Il software sviluppato segue delle fasi ben dettagliate,

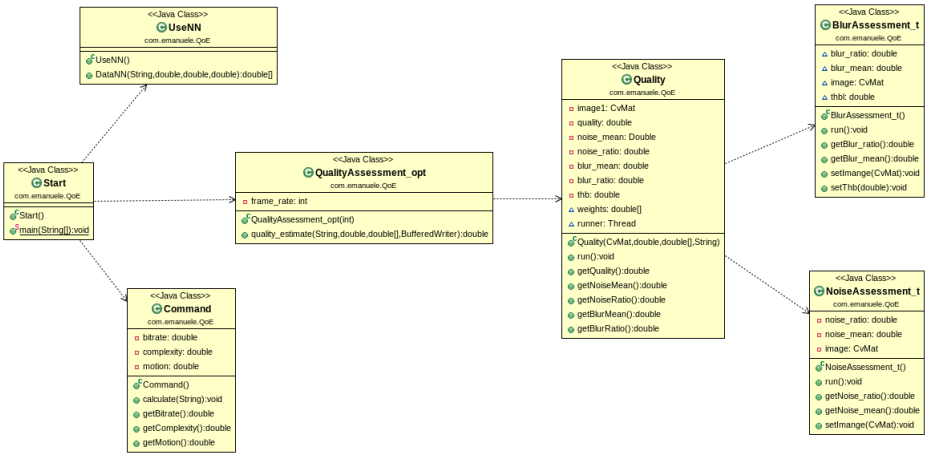


Figura 4.10: Diagramma delle classi implementate

come spiegato nel capitolo 3, che prevede l'estrazione delle caratteristiche del video dal bitstream, l'uso delle caratteristiche ottenute dalla ANN per avere le impostazioni ottimali del software per il video in esame, la cattura e l'elaborazione dei frame, la misura delle distorsioni che degradano la qualità, ed infine la stima della qualità complessiva. In questa sezione, verrà spiegato come sono state realizzate queste fasi dal punto di vista del codice e all'utilizzo delle librerie JavaCV e Neuroph.

### 4.3.1 Estrapolazione delle Features dal Bitstream

La prima operazione che effettua il *main*, classe *Start*, è quella di estrapolare le informazioni dal bitstream per calcolare la Complexity (C) e la Motion (M), che insieme alla Bitrate (B) saranno gli input della rete neurale artificiale ANN nella prossima fase. La classe che effettua l'analisi del bitstream e fornisce C, M, e B è la classe *Command*. Una volta istanziato l'oggetto viene richiamato il metodo contenuto nell'oggetto *calculate()*, che prende in ingresso il percorso del video da analizzare. Il metodo calculate esegue il comando ffmpeg come se si dovesse eseguire da linea di comando. Ffmpeg decodifica il video e restituisce tutte le informazioni contenute nel header del video, queste informazioni vengono convertite in stringhe e analizzate. Il metodo

calculate procede nel catturare in maniera diretta la B, che nel bitstream viene riportata in maniera esplicita, al valore viene applicata la formula 4.1 per la normalizzazione e salvata come un valore double nella variabile globale di tipo privato bitrate. Per quanto riguarda C e M queste devono essere calcolate, in quanto non sono riportate in maniera diretta. Perciò, vengono catturati i parametri di quantizzazione, $QP_I$, $QP_P$, dei frame I e P, e la grandezza dei frame I e P espressi in bit, $Bits_I$, $Bits_P$, in quanto questi dati sono accessibili direttamente dai dati ottenuti dal comando ffmpeg, e questi dati vengono applicate le equazioni per la stima di M e C, equazioni: 3.15, e anche questi valori vengono salvati come double nelle variabili globali di tipo privato complexity e motion, rispettivamente. La classe Command fornisce altri tre metodi, *getComlexity()*, *getMotion()* e *getBitrate()*, che come è facilmente comprensibile forniscono i tre valori calcolati necessari alla prossima fase.

### 4.3.2 Integrazione e uso della ANN

Una volta che il main ottiene i valori di Complexity, Motion e Bitrate, tramite i tre metodi definiti nella classe Command, procede alla stima della soglia del blur $Th_b$, fondamentale per decidere se un pixel è affetto da blur, e i pesi $w_{\mu_b}$, $w_{r_b}$, $w_{\mu_n}$, $w_{r_b}$, per definire quando blur e noise influenzano il degrado della qualità video. Come è stato già detto, questi parametri vengono decisi tramite l'utilizzo di una ANN. Come spiegato nella sezione lo sviluppo e l'applicazione della ANN è stato fatto tramite l'utilizzo del framework Neuroph. Il framework permette il salvataggio della ANN una volta finita la fase training e di test in un file nel formato .nnet. Al main viene dato in input il percorso del file che contiene la ANN. Si procede nel creare una istanza della classe *UseNN*, che sarà la classe che utilizzerà le API di Neuroph per l'interazione tra la ANN e codice Java. L'oggetto appena creato contiene il metodo *dataNN()*, che prende in ingresso il percorso del file della ANN, e tre valori double che consistono nella Complexity, Motion e Bitrate, calcolati nella fase precedente. Il metodo procede nel creare l'oggetto della classe *neuroph.core.NeuralNetwork*, utilizzando il costruttore della classe viene impostata la ANN fornendogli il percorso del file .nnet. L'oggetto rappresenterà lo strumento di interazione tra la ANN e il codice. Attraverso il metodo *setInput()*, contenuto nella classe *NeuralNetwork*, vengono impostati gli input, Complexity, Motion e Bitrate nel nostro caso, si procede nell'esecuzione del metodo *calculate()*, e la ANN calcolerà i parametri interessati. Usando il metodo *getOutput()*, tutti i dati di output verranno salvati in un array di double che verrà restituito al main attraverso il metodo *dataNN()*. Il main prima di procedere effettua dei controlli sui parametri restituiti per eventuali

errori riportati dalla ANN, precisamente se qualche valore è inferiore a 0.1, questo perché la ANN ha una piccola percentuale di errore e valori inferiori a 0.1 sono considerati troppo bassi, in quanto quasi azzererebbero la misura di blur e noise invalidando la stima complessiva della qualità. Alla fine di queste operazioni, il main possiede i valori ottimali della soglia del blur e dei pesi , in modo da poter settare le configurazioni ottimali del video in analisi nel core del software, cioè la parte di software che si occuperà della stima vera e propria.

### 4.3.3 Cattura ed Elaborazione dei Frame

Ottenuti i dati per adattare il software alle caratteristiche del video, si procede con la cattura dei frame, l'elaborazione dei frame la misura degli effetti distorsivi, per arrivare alla stima della qualità. La classe che dà il via a tutto questo processo è la classe *QualityAssessment*. Utilizzando il costruttore della classe vengono impostati: il video da analizzare, la soglia del blur $Th_b$, i pesi $w_{\mu_b}$, $w_{r_b}$, $w_{\mu_n}$, $w_{r_b}$, e il frame-rate. Il frame-rate, consiste in un parametro che regola ogni quanto frame si deve catturare un frame per essere analizzato. Quest'ultimo parametro, il frame-rate, influenza molto i tempi d'esecuzione del software. Una volta creato l'oggetto viene invocato il metodo *quality-estimate()* e si comincia la procedura di stima. Il primo passo consiste nella cattura del frame, per fare questo viene utilizzata una comoda classe contenuta in JavaCV, più precisamente in ffmpeg, *FfmpegFrameGrabber*, questa classe offre vari metodi per interagire con i dati dei video. Al costruttore della classe viene dato il percorso del video. Una volta creato l'oggetto viene utilizzato il metodo *setFormat()* per settare il formato del video, e il metodo *start()* per iniziare la sessione di scorrimento dei frame, entrambi i metodi sono contenuti nell'oggetto. Una volta che lo scorrimento dei frame viene iniziata, un altro metodo contenuto nell'oggetto creato dalla classe FfmpegFrameGrabber viene utilizzato per catturare i frame, il metodo *grab()*. Il metodo restituisce il frame come un oggetto della libreria OpenCV, *IplImage*. IplImage è una classe delle librerie OpenCV, che serve per dare una struttura adatta a contenere l'immagine. Catturato il primo frame, questo viene utilizzato per capire le dimensioni dei frame, altezza e larghezza, che serviranno dopo nell'elaborazione dei frame. Arrivati questo punto viene utilizzato un ciclo *while*, per eseguire il procedimento di stima su tutti i frame, finché il metodo grab() sarà diverso da null. All'interno del ciclo, in accordo con il frame-rate impostato all'inizio, si procede all'elaborazione del frame. Per prima cosa viene convertito il frame da colori a scala di grigi, in quanto la misura di blur e noise viene fatta con i valori di luminanza. La conversione viene fatta con un metodo offerto dalle librerie OpenCv, *cvC-*

*vtColor()*. Da qui in poi viene utilizzato un altra classe per il trattamento delle immagini, sempre una classe contenuta nelle librerie OpenCv, *CvMat*, che considera le immagini come delle matrici, e offre tanti metodi per il trattamento delle stesse immagini. Una volta ottenuto il frame in scala di grigi come un oggetto CvMat, si esegue una conversione dei valori dell'intensità luminosa dei pixel. I valori devono essere convertiti da un intervallo che varia tra 0 e 255 a un intervallo compreso tra 0 e 1, in quanto il metodo che sarà utilizzato per la stima del blur e del noise è stato realizzato per funzionare con valori di luminanza compresi tra 0 e 1, che sono anche dei valori standard per la rappresentazione della luminanza. Ultimata anche questa operazione, non resta che dividere l'immagine in quattro macro-blocchi, per effettuare le stime in parallelo utilizzando la tecnica di multithreading. I quattro macro-blocchi vengono creati con le stesse dimensioni attraverso il metodo *cvRect()* che restituisce una sotto-immagine dell'immagine originale, figura 4.11, sempre come un oggetto CvMat. Ottenuti i quattro macro-blocchi che verranno identificati con le etichette $A$, $B$, $C$ e $D$, si creano quattro thread della classe *Quality*, che effettueranno la stima del blur e del noise, e stimeranno la qualità per ogni macro-blocco.


Figura 4.11: Rappresentazione dei quattro Macro-Blocchi

### 4.3.4 Misura della Distorsione

La classe Quality è la classe che effettua la stima della qualità del macro-blocco. Essa è un thread e quindi implementa l'interfaccia *Runnable*, questo è il modo classico di creare un thread in java. Ogni thread di Quality, crea altri due thread che corrispondo a *BlurAssessment* per la misura del blur e *NoiseAssessment* per la misura del noise, questo per eseguire le stime delle

distorsioni in parallelo. Il processo prosegue creando gli oggetti di BlurAssessment e di NoiseAssessment, ad entrambi tramite il metodo *setImage()*, contenuto in entrambi le classi, viene data in input l'immagine da analizzare, che in questo caso è un macro-blocco, in più nell'oggetto di BlurAssessment viene impostata anche la soglia del blur attraverso il metodo *setThb()*. Infine viene creato il thread attraverso la classe Thread e vengono lanciati in parallelo con il metodo *start()*.

#### 4.3.4.1 Misura del blur

Anche la classe BlurAssessment implementa l'interfaccia Runnable. Questa classe offre i metodi *setImage()* per impostare l'immagine da esaminare, *getBlurMean()* e *getBlurRatio()* per restituire i valori calcolati con il processo di stima del blur, e *setThb()* per impostare la soglia del blur calcolata con la ANN. Per calcolare i valori di **BlurMean** e **BlurRatio** viene applicato l'algoritmo 1 riscritto in java. Per prima cosa attraverso l'uso dell'oggetto CvMat vengono create due matrici, *DiffHorVal* e *DiffVerVal*, che contengono i valori della differenza dei pixel in entrambe le direzioni. Vengono calcolate le medie orizzontali e verticali *D-H-Mean* e *D-V-Mean*. Si procede come descritto nell'algoritmo 1 calcolando gli edge, cioè, confrontando il valore dei pixel con le medie *D-H-Mean* per gli edge orizzontali e *D-V-Mean* per gli edge verticali. Per ogni pixel trovato che si trova in un edge viene incrementato un contatore *n-edge*. Vengono poi calcolati i valori del rapporto orizzontale e verticale per la decisione finale del blur per ogni pixel. Si applica la decisione finale ad ogni pixel, cioè, se il massimo di questi rapporti è minore della soglia $Th_b$, allora il pixel è affetto da blur, quindi si somma alla variabile *sum-max* il valore massimo tra i rapporti e si incrementata un contatore, *n-blur*, per i pixel corrotti dal blur. Infine, vengono stimati il BlurMean come il rapporto tra le variabili sum-blur e n-blur e il BlurRatio come rapporto tra n-blur e n-edge.

#### 4.3.4.2 Misura del noise

Analogamente anche la classe NoiseAssessment implementa l'interfaccia Runnable. Ha la stessa struttura della classe BlurAssessment con i metodi setImage(), getNoiseMean() e getNoiseRatio(). Per calcolare i valori di **NoiseMean** e **NoiseRatio** viene applicato l'algoritmo 2 riscritto in java. Si comincia applicando il filtro mediano offerto dalle librerie OpenCV, utilizzando il metodo *cvSmooth()* e impostando l'immagine originale, l'immagine di destinazione, il tipo di filtro da applicare, in questo caso *CV-Median*, e la grandezza del kernel da applicare a all'immagine, seguendo le indicazioni

degli autori viene impostato una grandezza del filtro 3x3. Si continua calcolando le matrici che contengono le differenze dei pixel adiacenti in entrambe le direzioni *diff-hor* e *diff-ver* e le loro medie *d-h-mean* e *d-v-mean*, sempre con l'uso dell'oggetto CvMat. Si procede come descritto nell'algoritmo 2, cioè confrontando il valore dei pixel con le medie d-h-mean e d-v-mean. Si confrontano i valori delle matrici diff-hor e diff-ver con le loro medie, e se risultano essere minori entrambi, viene inserito il valore massimo nella matrice *noise-p* altrimenti viene inserito il valore 0, questi pixel sono candidati a possibili pixel corrotti dal rumore. Viene calcolata la media *n-mean*. Si procede alla stima del rumore, confrontando i valori della matrice noise-p con la media n-mean, se il valore è più alto della media n-mean, allora il pixel è corrotto dal rumore, e ne viene incremento la variabile *n-count*, e vengono sommati i valori nella variabile *sum-noise*. Vengono calcolati NoiseRatio come il rapporto di n-count e il numero totale dei pixel e NoiseMean come il rapporto di sum-noise e n-count.

### 4.3.5 Calcolo della qualità complessiva

La classe Quality, una volta che ha lanciato in esecuzione i due thread per la stima del blur e del noise, resta in attesa che i due thread terminano le loro operazioni, attraverso il metodo *join()* offerto dalla classe *Thread*. Una volta ripreso il controllo la classe Quality usa i metodi getNoiseMean() e getNoiseRatio() della classe NoiseAssessment e getBlurMean() e getBlurRatio() della classe BlurAssessment. Salvando i valori nelle opportune variabili, applica la metrica descritta nell'equazione 3.1, utilizzando i pesi stimati con la ANN. Arrivati a questo punto è stata calcolata la qualità del macro-blocco, e qui il thread Quality finisce il suo compito. La classe QualityAssessment aspetta che i quattro thread $A$, $B$, $C$ e $D$ finiscano il loro compito, sempre attraverso il metodo join(), e ottiene i quattro valori della qualità utilizzando i metodi *getQuality()* contenuto nella classe *Quality*. I quattro valori vengono sommati e divisi per quattro ottenendo la qualità dell'intero frame. Per ogni frame da analizzare viene ripetuta questa operazione, le qualità stimate per ogni frame vengono sommate in una variabile per poi essere divise per il numero di frame esaminati, così infine viene stimata la qualità complessiva del video. Con il metodo quality-estimate() della classe QualityAssessment, si restituisce la qualità complessiva al main che la salverà in un file di testo, e qui finisce il ciclo di vita del software.

# Capitolo 5

# Risultati e Conclusioni

In questo capitolo verranno presentati i risultati delle sperimentazioni atte a dimostrare le proprietà ed il corretto funzionamento del modello di predizione della qualità video proposto, cercando di simulare la percezione della qualità dal sistema visivo umano. Al fine di ottenere dei risultati attendibili, per le simulazioni è stato utilizzato un emulatore di rete, in modo da poter testare il metodo proposto attraverso la ricreazione di reali scenari delle reti di telecomunicazioni. Di seguito vengono elencati i principali test eseguiti con i parametri utilizzati ed i relativi risultati.

## 5.1 Simulazioni per la valutazione della predizione della qualità

Per valutare il metodo proposto nei capitoli precedenti, si potevano implementare vari scenari in cui la qualità di un video streaming veniva degradata da fattori legati alla congestione della rete. Si poteva scegliere di condurre test attraverso la rete internet vera e propria o altre tipi di reti esistenti, ma vista l'enormità delle reti e dai molteplici fattori di degrado si è preferito testare il software implementato per via di simulazioni svolte in ambito controllato in laboratorio. Le simulazioni effettuate consistono nell'uso di un video server RTP, un client, entrambi collegati direttamente agli input e gli output di un emulatore di rete (PacketStorm Hurricane II), (figura 5.1). Il video server RTP è stato fornito da un vasto set di diversi tipi di video, e offre la loro visione attraverso l'avvio di una sessione streaming da parte del client. L'emulatore di rete utilizzato può generare vari tipi distorsioni come ritardo, jitter o perdita dei pacchetti, tutti in real-time e in accordo con i range dei modelli delle reti esistenti. Il tipo di client utilizzato per le simulazioni è un laptop Asus X53S con sistema operativo Ubuntu 10.14 e le

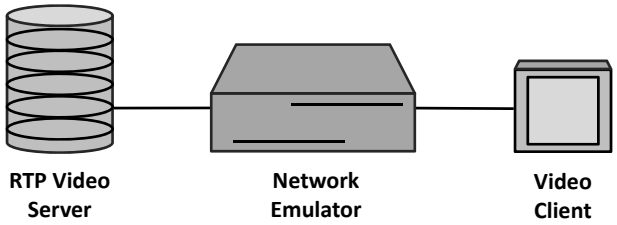


Figura 5.1: Modello utilizzato per le simulazioni

seguenti caratteristiche hardware: processore intel core i7 2630QM, scheda grafica nvidia GeForce GT 520MX 1Gb e 8Gb di memoria RAM. Ai fini dei risultati, nel client sono stati lancianti in esecuzione il metodo sviluppato e proposto, e la metrica FR SSIM con lo scopo di realizzare una analisi comparativa dei risultati ottenuti con entrambi i metodi. Mentre il SSIM è ben accettato come riferimento delle performance del QoE, l'approccio del metodo presentato nei precedenti capitoli è funzionalmente più applicabile in sistemi streaming realistici, in quanto non necessita avere il video originale (senza distorsioni) attraverso la rete utilizzata per lo streaming del video dal lato client, cosa indispensabile per utilizzare la metrica SSIM. In primo luogo il client si connetterà al server, il server fornirà al client il modello della ANN precedentemente allenata, come mostrato nella sezione 4.2.2, con il training set costruito, (sezione 4.2.1). Ultimato questo processo, il client è pronto per la fase di stima della qualità.

## 5.2 Configurazioni delle simulazioni

Durante la sessione di streaming che il client avvia, viene cominciata la trasmissione del video da parte del server. Mentre lo streaming prende piede, il metodo sviluppato comincia l'analisi dei frame in real-time (senza innescare in nessun modo ritardo al processo di streaming). Dovuto alle caratteristiche hardware del client, l'algoritmo impiega 0.3 secondi a stimare la qualità di un frame. Per questo motivo è stato introdotto il frame rate come parametro da impostare in base al tipo di dispositivo del client. In queste simulazioni, tenendo conto delle caratteristiche hardware del dispostivo client è stato impostato il frame rate a 8, cioè, viene catturato e processato un frame ogni otto frame, che per i video, con stream a 24 fps, utilizzati nelle simulazioni e nelle fasi precedenti, corrispondono a 3 frame per secondo. Qualora il dispositivo del client fosse un dispositivo più leggero, come ad esempio uno smart-phone o un tablet, per rispettare la condizione di real-time può essere abbassato il valore del frame rate. Una volta finito lo streaming del video, viene salvata

una copia dell'intero video con le distorsioni introdotte dal canale di comunicazioni, per essere poi utilizzato con la metrica FR SSIM insieme al video originale, che il client già dispone, al fine di comparare i risultati con entrambi i metodi. L'emulatore di rete utilizzato, è in grado di generare diversi tipi di disturbi (es. ritardo, jitter o perdita di pacchetti) in real-time. È stato dimostrato che il fattore della perdita dei pacchetti degrada la qualità più di ogni altro fattore [46]. Per tale motivo, tutte le simulazioni sono state fatte utilizzando questo fattore di degrado, avviando la prima sessione di streaming senza nessuna perdita di pacchetto. Poi il processo di streaming veniva ripetuto, aumentando di grado in grado la percentuale di pacchetti persi, da 0,5%, 1%, 5%, 10% per arrivare infine a 20%, rispettivamente. Una volta finita la sessione, il client accede al video originale con lo scopo di utilizzare la metrica SSIM per l'analisi comparativa di benchmarking. Da notare che i parametri ricavati dalla ANN sono ottenuti da un processo di training su video non distorti dalla trasmissione (Figura 4.6). Gli esperimenti sono invece stati eseguiti su video distorti dall'emulatore di rete. In tal modo si è cercato di validare l'accuratezza del nostro metodo in un caso estremo. Più realistico sarebbe un processo in cui il training della ANN include anche video distorti.

## 5.3 Risultati ottenuti

Le figure 5.2 e 5.3, mostrano i risultati dei test effettuati con i video tr1 e sh1 codificati con bitrate 2048 kbps. Le linee rosse rappresentano i risultati ottenuti con il metodo sviluppato, le linee blu i risultati di benchmarking (SSIM) e le linee verdi i risultati ottenuti con il metodo senza l'utilizzo della ANN. Le differenti caratteristiche in termini di complexity (C) e motion (M) di questi due video, valori bassi per sh1 (0.16 e 0.18) e valori molto alti per tr1 (0.33 e 0.25), li rendono dei candidati perfetti per valutare a fondo le performance del metodo presentato sotto condizione estreme. In entrambi i casi, l'indice di qualità video calcolata dal metodo proposto (linea rossa) si degrada con l'aumentare del livello di pacchetti persi, partendo approssimativamente da 0,98 (quando la percentuale di pacchetti persi è settata a 0) arrivando a 0,75 per tr1 e 0.64 per sh1 (quando la percentuale di pacchetti persi è settata a 20). La differenza della degradazione della qualità tra i video testati è giustificata dal fatto che i video hanno dinamiche molto diverse tra loro (valori di M e C diversi tra loro).

Con la perdita di pacchetti si degrada la qualità e C ed M hanno una grande influenza, in quanto cambiando i valori di pacchetti persi, C e M variano sensibilmente. Questo provoca che per video codificati con la stessa bitrate ma con bassi livelli di C e M si degradano più velocemente, cioè hanno

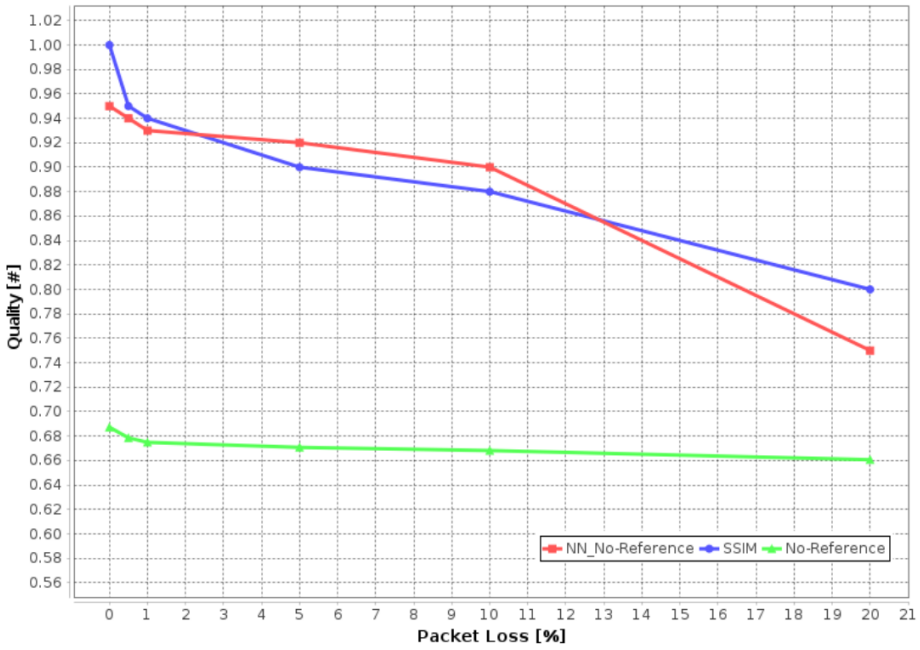


Figura 5.2: Risultati relativi al video tr1 codificati a 2048 kbps, per differenti valori di pacchetti persi.

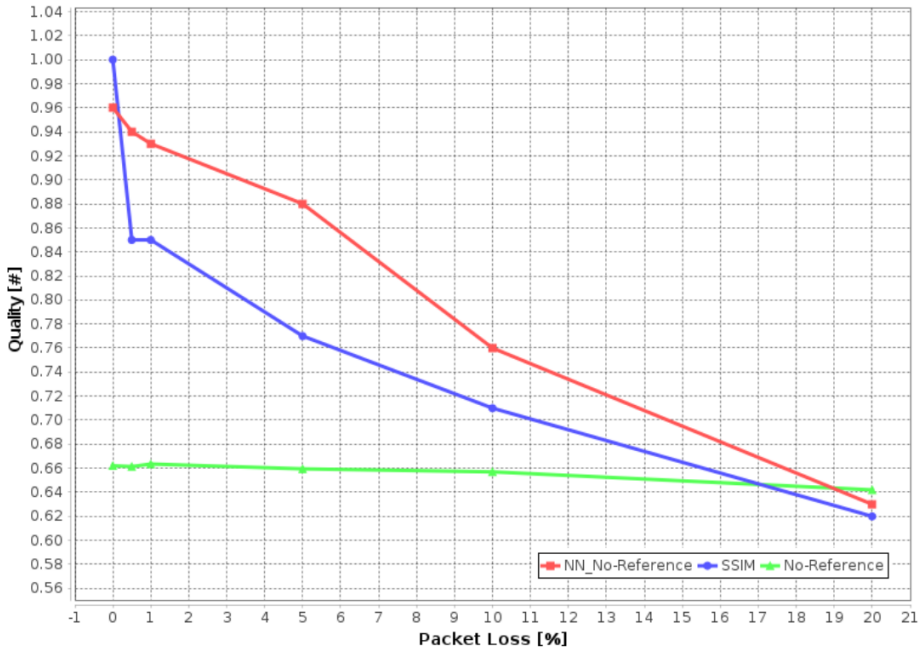


Figura 5.3: Risultati relativi al video sh1 codificati a 2048 kbps, per differenti valori di pacchetti persi.

una qualità più bassa rispetto ai video con valori di C e M più alti.

Nelle tabelle 5.1 e 5.2, vengono riportati i valori di C e M al variare del tasso di pacchetti persi dei rispettivi due video presi in considerazione per le simulazioni tr1 e sh1 codificati con una bitrate di 2048 kbps. Come si può vedere, i valori cambiano sensibilmente, specialmente la C di sh1, in quanto durante le simulazioni questo valore si mantiene molto basso, rispetto anche alla M di sh1. Per tale motivo, cioè, la grande differenza tra i valori di C e M del video sh1 rispetto alla differenza dei valori di tr1 che è minore, la qualità di sh1 degrada più sensibilmente rispetto alla qualità di tr1 fino ad arrivare a valori inferiori. Le tabelle 5.3 e 5.4 riportano i valori di $Th_b$ e $w_{\mu_b}$, $w_{r_b}$, $w_{\mu_n}$, $w_{r_b}$ dei due video. Questi valori, che vengono calcolati con l'uso di C e M attraverso una ANN, sono molto diversi tra i due video a causa ovviamente della diversità dei valori di C e M. I valori relativi a sh1 tendono ad essere più elevati dei valori di tr1, significando che blur e noise influenzano maggiormente il video sh1, e per tale motivo il valore della qualità restituito dal metodo è più basso rispetto a tr1, come si può notare dalle figure 5.2 e 5.3 (linea rossa).

| Video_% packet loss | Complexity | Motion |
|---|---|---|
| tr1_0% | 0.33 | 0.25 |
| tr1_0.5% | 0.262 | 0.207 |
| tr1_1% | 0.174 | 0.204 |
| tr1_5% | 0.191 | 0.21 |
| tr1_10% | 0.132 | 0.206 |
| tr1_20% | 0.05 | 0.185 |

Tabella 5.1: packet loss, complexity e motion tr1 codificato a 2048 kbps

| Video_% packet loss | Complexity | Motion |
|---|---|---|
| sh1_0% | 0.33 | 0.25 |
| sh1_0.5% | 0,05 | 0,194 |
| sh1_1% | 0.002 | 0.196 |
| sh1_5% | 0.08 | 0.196 |
| sh1_10% | 0.07 | 0.191 |
| sh1_20% | 0.05 | 0.151 |

Tabella 5.2: packet loss, complexity e motion sh1 codificato a 2048 kbps

| Video_% packet loss | $Th_b$ | $w_{\mu_b}$ | $w_{r_b}$ | $w_{\mu_n}$ | $w_{r_b}$ |
|---|---|---|---|---|---|
| tr1_0% | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| tr1_0.5% | 0.156 | 0.1 | 0.178 | 0.168 | 0.128 |
| tr1_0.5% | 0.156 | 0.1 | 0.178 | 0.168 | 0.128 |
| tr1_0.5% | 0.156 | 0.1 | 0.178 | 0.168 | 0.128 |
| tr1_0.5% | 0.156 | 0.1 | 0.178 | 0.168 | 0.128 |
| tr1_20% | 0.32 | 0.1 | 0.1 | 0.443 | 0.1 |

Tabella 5.3: valori della soglia $Th_b$ e dei pesi $w_{\mu_b}$, $w_{r_b}$, $w_{\mu_n}$, $w_{r_b}$, del video tr1 codificato a 2048 kbps

| Video_% packet loss | $Th_b$ | $w_{\mu_b}$ | $w_{r_b}$ | $w_{\mu_n}$ | $w_{r_b}$ |
|---|---|---|---|---|---|
| tr1_0% | 0.2 | 0.1 | 0.1 | 0.1 | 0.1 |
| tr1_0.5% | 0.214 | 0.121 | 0.228 | 0.3 | 0.08 |
| tr1_1% | 0.332 | 0.1 | 0.1 | 0.445 | 0.1 |
| tr1_5% | 0.5 | 0.125 | 0.231 | 0.3 | 0.2 |
| tr1_10% | 0.7 | 0.125 | 0.232 | 0.297 | 0.5 |
| tr1_20% | 0.9 | 0.1 | 0.1 | 0.277 | 0.8 |

Tabella 5.4: valori della soglia $Th_b$ e dei pesi $w_{\mu_b}$, $w_{r_b}$, $w_{\mu_n}$, $w_{r_b}$, del video sh1 codificato a 2048 kbps

Comparando i risultati del metodo proposto con i risultati del metodo FR SSIM (linea blu), si può osservare che sebbene in alcuni punti la qualità misurata è differente, dovuto al fatto che il metodo SSIM si concentra sulla somiglianza delle strutture mentre il metodo sviluppato si basa sull'influenza della complexity e della motion, il metodo è ben correlato con il metodo preso come punto di riferimento SSIM.
A dare prova di ciò vengono calcolate le correlazioni Pearson[1] e Spearman[2] così come RMSE[3] (errore quadratico medio) per quantizzare la correlazione tra i due metodi, tabella 5.5. Per entrambi i video, la correlazione Pearson e Spearman ha valori molto alti, maggiori di 0.9 e il RMSE restituisce valori inferiori a 0,08. Questi alti valori delle correlazioni indicano che il metodo sviluppato NR ha un comportamento in linea con il metodo preso come riferimento FR.
Un altro importante elemento che è stato esaminato è stabilire l'efficacia

| Correlazione | tr1 | sh1 |
|---|---|---|
| Pearson | | |
| SSIM-NR ANN | 0.9073 | 0.9073 |
| SSIM-NR no ANN | 0.8472 | 0.8008 |
| Spearman | | |
| SSIM-NR ANN | 1 | 0.9856 |
| SSIM-NR no ANN | 0.91 | 0.8986 |
| RMSE | | |
| SSIMvsNR-ANN | 0.0316 | 0.0716 |
| SSIMvsNR-noANN | 0.2433 | 0.1826 |

Tabella 5.5: Valori di correlazione tra SSIM e il metodo sviluppato con e senza l'uso della ANN per i video tr1 e sh1.

della ANN per il calcolo in automatico dei parametri utilizzati dall'algoritmo per la misura del QoE. Stabilire se realmente desse un effettivo miglioramento ai risultati. A questo fine è stato implementato un nuovo ciclo di simulazioni con le stesse caratteristiche delle precedenti, solo che non viene utilizzata la ANN (linea verde nelle figure 5.2 e 5.3), e i parametri necessari all'algoritmo vengono calcolati in maniera statistica, non tenendo conto della tipologia di

[1] In statistica, l'indice di correlazione di Pearson tra due variabili statistiche è un indice che esprime una eventuale relazione di linearità tra esse.

[2] è una misura statistica non parametrica di correlazione. Essa misura il grado di relazione tra due variabili per le quali non si fa altra ipotesi della misura ordinale, ma possibilmente continua.

[3] indica la discrepanza quadratica media fra i valori dei dati osservati ed i valori dei dati stimati.

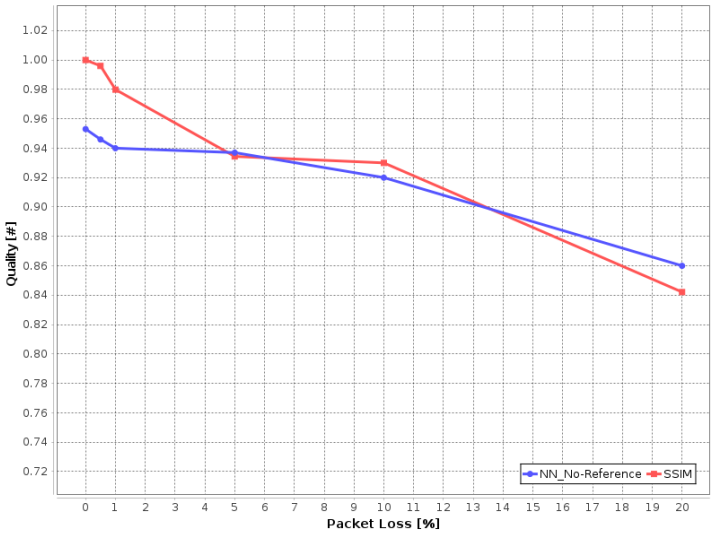


Figura 5.4: Risultati relativi al video pa1 codificati a 2048 kbps, per differenti valori di pacchetti persi.

video. I parametri sono calcolati in maniera univoca e sono uguali per tutti i tipi di video. Possiamo vedere (tabella 5.5) che la correlazione per entrambi i casi è diminuita di almeno 10%. L'aumento che si ha per quanto riguarda il RMSE è anche più estremo, raggiungendo valori di 0,2433 e 0,1826 per tr1 e sh1, rispettivamente.
Nelle figure 5.4, 5.5, 5.6, 5.7, 5.8, 5.9 vengono riportati i risultati del metodo proposto confrontati con il benchmark SSIM, di tutti i restanti video con bitrate 2048 a Kbps.
Da questi risultati, si conclude che il metodo ibrido sviluppato, rafforzato con l'uso di una ANN, è capace di stimare la qualità con risultati comparabili con lo stato dell'arte per le metriche FR. Il metodo permette la misura della qualità in tempo reale e quindi un analisi del video in real-time, grazie alla complessità ridotta e non ha bisogno del materiale originale. Inoltre, grazie alla sua bassa complessità e tempo di esecuzione può essere eseguito in dispositivi con risorse abbastanza limitate come i dispositivi mobile.

## 5.4 Conclusioni

In questo elaborato è stato presentato un nuovo metodo No-Reference per la stima della qualità video analizzando frame per frame il video. Più esattamente, basandosi su una valutazione lineare tra blur e noise per la stima della qualità nelle immagini, è stato disegnato e implementato un metodo a bassa

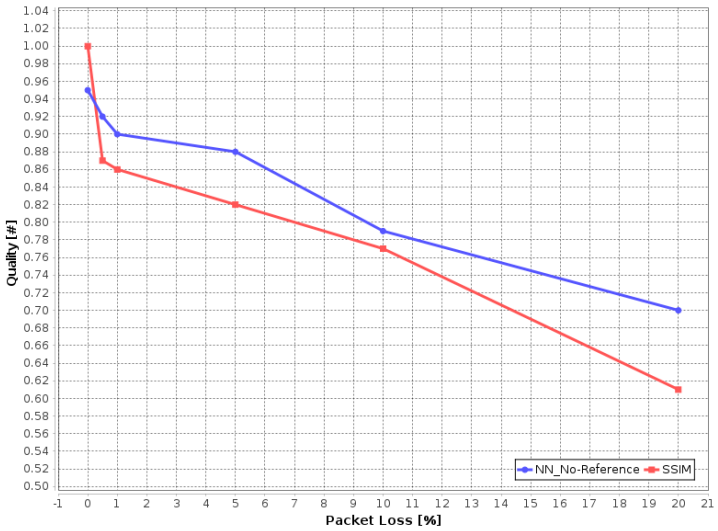


Figura 5.5: Risultati relativi al video pr1 codificati a 2048 kbps, per differenti valori di pacchetti persi.

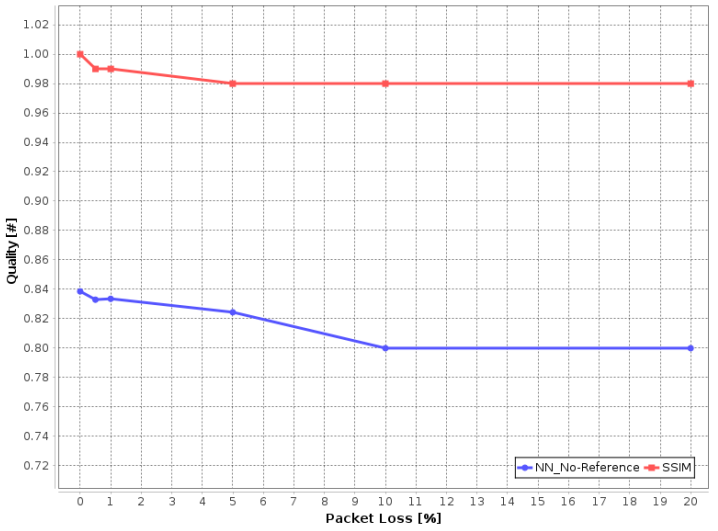


Figura 5.6: Risultati relativi al video rb1 codificati a 2048 kbps, per differenti valori di pacchetti persi.

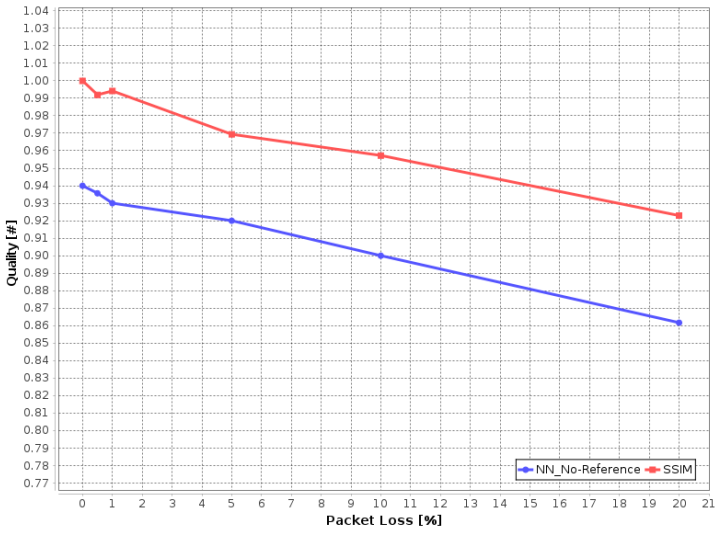


Figura 5.7: Risultati relativi al video rh1 codificati a 2048 kbps, per differenti valori di pacchetti persi.

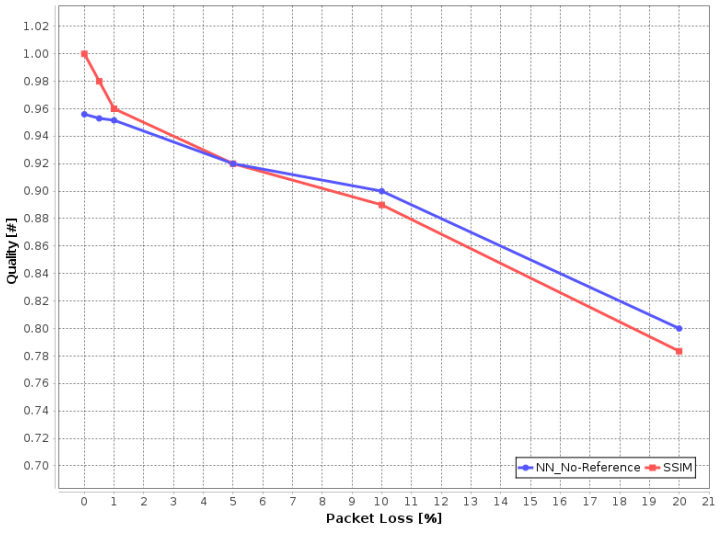


Figura 5.8: Risultati relativi al video sf1 codificati a 2048 kbps, per differenti valori di pacchetti persi.

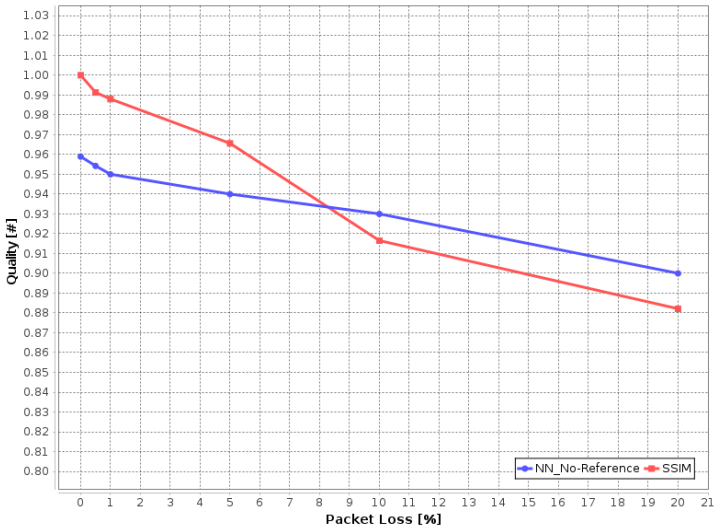


Figura 5.9: Risultati relativi al video st1 codificati a 2048 kbps, per differenti valori di pacchetti persi.

complessità NR per la stima della qualità dei video, adattato per funzionare in real-time e per la valutazione end-to-end delle performance dei servizi. Una Rete Neurale Artificiale è stata utilizzata per adattare automaticamente i pesi del blur e del noise (quanto blur e noise influenzano quel tipo di video) alle diverse caratteristiche del video, rendendo l'algoritmo dinamico, capace di adattarsi ad una vasta gamma di video.
Quello che si è cercato di fare è di creare una nuova metrica oggettiva ben correlata con il sistema visivo umano (HVS), che riuscisse a quantizzare la qualità percepita dall'uomo. Per tale motivo il metodo è stato comparato con la metrica FR SSIM, considerata lo stato dell'arte in quanto essere ben correlata con il HVS, ma non utilizzabile in un contesto online, data la necessità obbligatoria del video originale per effetuare la stima. In un ambiente di rete emulata è stato valutato il metodo presentato NR con il metodo alternativo FR SSIM, trovando una forte correlazione (RMSE minore di 0,07 in tutti i scenari provati). Le simulazioni sono state eseguite creando scenari di distorsione del video, variando la percentuale di pacchetti persi. Altre tipi di distorsioni potevano essere prese in considerazioni, ma è stato considerato il fattore di pacchetti persi come il più rilevante per il degrado della qualità video. Data la bassa complessità computazionale del metodo, si può dire che il software sia adatto per funzionare su dispositivi mobile con risorse limitate (es. smart-phone o tablet), in reali reti mobile (come 4G-LTE) e in pratici sistemi di fornitura di servizi (dove i video originali non sono disponibili dal

lato client).

## 5.5 Sviluppi futuri

Il metodo presentato ha dimostrato di avere buoni risultati durante le fasi di testing e si potrebbe pensare di utilizzarlo direttamente su una rete reale, ma prima bisognerebbe effettuare delle considerazioni. Una prima considerazione è che in questo elaborato si è pensato di assumere che l'effetto blur è uno dei maggiori fattori di degrado della qualità. Questo è corretto per le immagini, in quanto durante la fase di compressione può capitare che si elimini qualche frequenza e ciò introdurrebbe blur all'immagine. Per quanto riguarda i video trasmessi con perdita di pacchetto il blur non ha un effetto immediato, cioè non esiste una relazione lineare tra perdita di pacchetto e blur, e quindi cercare di calcolare la quantità di blur introdotta a causa della perdita di pacchetti può fallire per alcuni versi risentendone di accuratezza dei risultati. Per migliorare la stima della quantità di blur si potrebbe estendere il calcolo non al singolo frame ma ad un gruppo di frame e stimare il blur esteso per il gruppo di frame in un intervallo di tempo, in modo da poter analizzare la distorsione che il pacchetto perso introduce su tutto il gruppo di frame. Inoltre, la perdita di pacchetti potrebbe influenzare più direttamente altri tipi di artefatti, quindi il metodo si potrebbe estendere al calcolo di nuovi tipi di distorsione al fine di migliorare l'accuratezza.
Altra considerazione riguarda la *Artificial Neural Network*. La costruzione della ANN adoperata è stata eseguita in maniera euristica, come anche la determinazione dei parametri di learning. Il risultato è stato una semplice struttura della ANN con una bassa percentuale di errore, ma un ulteriore aggiornamento con una modifica dei parametri di learning potrebbe portare ad avere da una parte una struttura della ANN più complessa, ma dall'altra un miglioramento dei risultati con la conseguenza di un abbassamento della percentuale di errore. Ciò comporterebbe alla ANN di trovare una migliore correlazione tra gli input *Complexity*, *Motion* e *Bitrate* e gli output soglia blur ($Th_b$) e pesi ($w_{\mu_b}$, $w_{r_b}$, $w_{\mu_n}$, $w_{r_b}$), in quanto tali parametri hanno una complicata relazione non lineare non semplice da ricostruire.
Come detto la costruzione del training set con la quale è stata allenata la ANN è stato formato con i 100 video forniti dal database LIVE. Quindi, sono stati utilizzati solamente i video originali con codifica MPEG-4 AVC/H.264 per allenare la rete, di conseguenza la ANN utilizzata non ha conoscenza sul comportamento dei video degradati. Gli output che si ottengono vengono stimati tramite una associazione ai video originali e questo può portare ad una perdita di accuratezza. Al fine di migliorare i risultati si potrebbe integrare

al training set i dati dei video degradati durante la trasmissione, in modo che la ANN prendesse anche in considerazione i video degradati per affinare la correlazione necessaria allo scopo.

# Bibliografia


[1] Cisco Visual Networking Index, Global mobile data traffic forecast update, 2013-2018. Cisco white paper (2014).

[2] M. Torres Vega, S. Zou, and D. C. Mocanu, E. Tangdiongga, A. M. J. Koonen, and A. Liotta, *End-to-End Performance Evaluation in High-Speed Wireless Networks*. The Int. Conf. on Network and Service Management (CNSM), 2014.

[3] M. Shahid, A. Rossholm, B. Lövström, and H. Zepernick, *No-reference image and video quality assessment: a classification and review of recent approaches*. Journal on Image and Video Processing, vol. 40, no. 1, 2014.

[4] ITU, ITU-R Recommendation BT.500-13 *Methodology for the subjective assessment of the quality of the television pictures*. http://www.itu.int/rec/R-REC-BT.500-13-201201-I/en. Accessed 4 November 2013

[5] HR. Wu, KR. Rao, *Digital Video Image Quality and Perceptual Coding (Signal Processing and Communications)*. (CRC, Boca Raton, 2005)

[6] MG. Choi, JH. Jung, JW. Jeon, *No-reference image quality assessment using blur and noise*. Int. J. Comput. Sci. Eng. 3(2), 76–80 (2009)

[7] ITU, ITU-R Recommendation P.800 *Methods for subjective determination of transmission quality*. http://www.itu.int/rec/T-REC-P.800-199608-I/en.

[8] P. Juluri, V. Tamarapalli, D. Medhi *Measurement of Quality of Experience of Video-on-Demand Services: A Survey*. DOI 10.1109/COMST.2015.2401424, IEEE Communications Surveys  Tutorials

[9] T. Hoßfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, C. Lorentzen, *Initial delay vs. interruptions: Between the devil and the deep blue sea*. in Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on, July 2012, pp. 1–6.

[10] B. Lewcio, B. Belmudez, T. Enghardt, S. Moller, *On the way to high-quality video calls in future mobile networks*. in Proc. Quality of Multimedia Experience (QoMEX), 2011, pp. 43–48.

[11] M. Montenovo, A. Perot, M. Carli, P. Cicchetti, A. Neri, *Objective quality evaluation of video services*. Proceedings of the Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics., 2006.

[12] S. Winkler, A. Sharma, D. McNally, *Perceptual video quality and blockiness metrics for multimedia streaming applications*. in Proceedings of the International Symposium on Wireless Personal Multimedia Communications, 2001, pp. 547–552.

[13] L. Yitong, S. Yun, M. Yinian, L. Jing, L. Qi, Y. Dacheng, *A study on quality of experience for adaptive streaming service*. in Communications Workshops (ICC), 2013 IEEE International Conference on, June 2013, pp. 682–686.

[14] M. Fiedler, T. Hoßfeld, P. Tran-Gia, *A generic quantitative relationship between quality of experience and quality of service*. Network, IEEE, vol. 24, no. 2, pp. 36–41, 2010.

[15] S. Winkler, P. Mohandas, *The evolution of video quality measurement: From PSNR to hybrid metrics*. IEEE Transactions on Broadcasting, vol. 54, no. 3, pp. 660–668, Sep. 2008.

[16] Q. Huynh-Thu, M. Ghanbari, *Scope of validity of PSNR in image/video quality assessment*. Electronics Letters 44 (13): 800. doi:10.1049/el:20080522

[17] E. Oriani, *qpsnr: A quick PSNR/SSIM analyzer for Linux*. Retrieved 6 April 2011.

[18] *pnmpsnr User Manual*. Retrieved 6 April 2011.

[19] Z. Wang, L. Lu, A. C. Bovik, *Video quality assessment based on structural distortion measurement*. Signal Processing: Image Communication, vol. 19, no. 2, pp. 121–132, Feb. 2004.

[20] R. Dosselmann, X. D. Yang *A comprehensive assessment of the structural similarity index*, page-1, DOI 10.1007/s11760-009-0144-1

[21] S. Hemami, A. Reibman, *No-reference image and video quality estimation: applications and human-motivated design*. Signal Process. Image Commun. 25(7), 469–481 (2010)

[22] M. Shahid, A. Rossholm et al. *No-reference image and video quality assessment: a classification and review of recent approaches*. EURASIP Journal on Image and Video Processing 2014 2014:40.

[23] L. Liang, S. Wang, J. Chen, S. Ma, D. Zhao, W. Gao, *No-reference perceptual image quality metric using gradient profiles for JPEG2000*. Signal Process. Image Commun. 25(7), 502–516 (2010)

[24] S. Winkler, *Digital Video Quality: Vision Models and Metrics*. (Wiley, Chichester, 2005)

[25] N. Narvekar, L. Karam, *A no-reference image blur metric based on the cumulative probability of blur detection (CPBD)*. IEEE Trans. Image Process. 20(9), 2678–2683 (2011)

[26] A. Bovik, *Handbook of Image and Video Processing*, 2nd edn. Elsevier Academic, Burlington, 2005

[27] A. Amer, E. Dubois, *Fast and reliable structure-oriented video noise estimation*. IEEE Trans. Circuits Syst. Video Technol. 15(1), 113-118 (2005)

[28] K. Rank, M. Lendl, R. Unbehauen, *Estimation of image noise variance*. IEEE Proc. Vis. Image Signal Process. 146(2), 80–84 (1999)

[29] H. Wu, K. Rao, *Digital Video Image Quality and Perceptual Coding*. Signal Processing and Communications CRC, Boca Raton, 2005

[30] A. Reibman, V. Vaishampayan, Y. Sermadevi, *Quality monitoring of video over a packet network*. IEEE Trans. Multimedia 6(2), 327–334 (2004)

[31] A. Takahashi, D. Hands, V. Barriac, *Standardization activities in the ITU for a QoE assessment of IPTV*. IEEE Commun. Mag. 46(2), 78–84 (2008)

[32] A. Raake, M. Garcia, S. Moller, J. Berger, F. Kling, P. List, J. Johann, C. Heidemann, *T-V-model: parameter-based prediction of IPTV quality*, in IEEE International Conference on Acoustics, Speech and Signal Processing (Las Vegas, 31 March to 4 April 2008), pp. 1149–1152

[33] J. Han, Y. Kim, J. Jeong, J. Shin, *Video quality estimation for packet loss based on no-reference method*, in IEEE International Conference on Advanced Communication Technology, vol. 1 (Phoenix Park, Dublin, 7–10 February 2010), pp. 418–421

[34] T. Yamada, S. Yachida, Y. Senda, M. Serizawa, *Accurate video-quality estimation without video decoding*, in IEEE International Conference on Acoustics Speech and Signal Processing (Dallas, 14–19 March 2010), pp. 2426–2429

[35] A. Rossholm, B. Lövström, *A new low complex reference free video quality predictor*, in IEEE Workshop on Multimedia Signal Processing (Cairns, 8–10 October 2008), pp. 765–768

[36] M. Naccari, M. Tagliasacchi, S. Tubaro, *No-reference video quality monitoring for H.264/AVC coded video*. IEEE Trans. Multimedia 11(5), 932–946 (2009)

[37] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, *Image quality assessment: from error visibility to structural similarity*. IEEE Trans. Image Process. 13(4), 600–612 (2004)

[38] A. Ichigaya, Y. Nishida, E. Nakasu, *Non reference method for estimating PSNR of MPEG-2 coded video by using DCT coefficients and picture energy*. IEEE Trans. Circuits Syst. Video Technol. 18(6), 817–826 (2008)

[39] A. Liotta, D. C. Mocanu, V. Menkovski, L. Cagnetta, G. Exarchakos, *Instantaneous video quality assessment for lightweight devices*. in Proc. of Int. Conference on Advances in Mobile Computing, ser. MoMM '13, New York, NY, USA, 2013, pp. 525:525–525:531.

[40] J. Hu, H. Wildfeuer, *Use of content complexity factors in video over ip quality monitoring*. in Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on, 2009, pp. 216–221.

[41] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Washington DC, USA: Spartan Books, 1961.

[42] *Neuroph, Java Neural Network Framework.* http://neuroph.sourceforge.net/.

[43] *JavaCV, Java interface to OpenCV and more.* https://github.com/bytedeco/javacv.

[44] *The missing bridge between Java and native C++*. https://github.com/bytedeco/javacpp.

[45] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, L. K. Cormack, *Study of Subjective and Objective Quality Assessment of Video*, IEEE Transactions on Image Processing , vol.19, no.6, pp.1427-1441, June 2010.

[46] D. Mocanu, A. Liotta, A. Ricci, M. Vega, and G. Exarchakos, *When does lower bitrate give higher quality in modern video services?*, in Network Operations and Management Symposium (NOMS), 2014 IEEE, May 2014, pp. 1–5.

[47] 7th International Workshop on Quality of Multimedia Experience 26-29 May 2015 Costa Navarino, Messinia, Greece. http://www.qomex.org/

[48] M. Mohri, A. Rostamizadeh, A. Talwalkar (2012). *Foundations of Machine Learning*, The MIT Press. ISBN 9780262018258.

# Ringraziamenti

Oggi 20 aprile del 2015 finisce la mia bellissima vita da universitario, sono passati tanti anni dal lontano settembre 2006, ricordo quel giorno come se fosse ieri, l'inizio della mia vita universitaria. Fresco di diploma (62/100) e di mondiali vinti dall'Italia, attraversavo lo stretto di Messina deciso a consegnare la domanda di immatricolazione al corso di laurea in Ingegneria Informatica. Ricordo che neanche la riuscivo a trovare l'entrata della facoltà, tant'è che dovetti scavalcare un cancello per raggiungere la segreteria e consegnare la domanda, forse questo era un monito, chi lo sa. Sta di fatto che qui oggi a Padova finisce, iniziata al sud finita al nord, storie di tanti disgraziati come me. Ma sopratutto iniziata che confondevo le parentesi graffe con gli integrali finita con una pubblicazione ad una conferenza della IEEE. Ma la cosa più importante per me, è la cosa che c'è stata in mezzo a questi due opposti, tutte le fantastiche persone che hanno reso questa avventura bellissima e che mi sento in obbligo di ringraziare, sarà certamente difficile elencarle tutte, ma ci proverò lo stesso, scusandomi da subito con chi non è stato menzionato.

Questo lavoro di tesi e la pubblicazione non sarebbero stati possibili senza il Prof. Antonio Liotta, la ringrazio professore per la fiducia e per avermi dato la possibilità di lavorare per lei ad Eindhoven, nonostante il mio livello d'inglese. Ringrazio anche Maria e Dec, grazie per il sostegno e per l'aiuto, è stato un piacere lavorare con voi.

Ma ancora prima un ringraziamento speciale va a mia sorella Luana. La vera autrice del mio erasmus in Olanda, grazie a lei ho conosciuto il Prof. Liotta che poi ha dato il via a tutto, ma la ringrazio sopratutto per l'affetto che solo lei può darmi e per le grandi risate che ci siamo fatti insieme a nostra madre.

Speciale il ringraziamento per mia mamma Teresa, donna fantastica di grande tenacia. Tu sei stata ad insegnarmi la vita e ad essere furbi, a ridere sempre a tutto e a non buttarsi mai a terra.

Ringrazio mio fratello Danilo e mio papà Domenico (Mimmo), per avermi sempre sostenuto ed avermi dato la possibilità di essere dove sono adesso.

Insomma ringrazio tutta la mia famiglia, che con moltissimi sacrifici mi ha dato la possibilità di studiare a Padova, spero di non avervi mai deluso e di avervi resi orgogliosi di me.
Ringrazio i miei amici calabresi, amici di sempre, che ogni tanto riusciamo ancora a radunarci tutti insieme, Luca, Davide, Gianni, Rocco, Gimmy, Gianluca (sukina), i grandi PPJ. Quante avventure passate insieme. Indimenticabili quei 4 mesi di convivenza a Messina. Convivenza che è durata solo con Gianni a lui va un ringraziamento particolare, 5 anni di convivenza a Messina, indimenticabile in due sul motorino contro un uragano di pioggia e vento cantando la sigla di Street Fighter per raggiungere l'università.
Ringrazio Marco, Roberta, Rosaria, Ivan, Luca, Daniela e tutti gli atri mie colleghi di Messina, con voi ho cominciato questa esperienza, ogni tanto riguardo le vecchie foto e ricordi tornano a salire, le prime lezioni tutti insieme e i pomeriggi a studiare in università, le innumerevoli risate. Ringrazio specialmente a Marco e Ivan per essere venuti a trovarmi ad Eindhoven, spero di avervi fatto divertire. Speriamo di restare sempre in contatto con tutti e di rivederci presto tutti insieme.
Ringrazio tutti i miei infiniti cugini, ma sopratutto Gianluca (sukina) e Daniele che con voi sono cresciuto e tante cose abbiamo fatto insieme. Altri cugini da ringraziare, che sono stati il mio ponte tra Taurianova e Padova, sono Graziano e Peppe, grazie per tutto il sostegno che mi avete dato, ma sopratutto a Graziano che mi ha fatto conoscere una realtà bellissima che ho vissuto in questi ultimi tre anni, il Collegio Don Mazza.
Desidero ringraziare tutto il Collegio Don Mazza, Don Mario, Don Flavio, Gianluca, Mirco, Patrizia e tutti gli atri, se oggi sono qui a Padova lo devo anche a voi, ringrazio specialmente Don Mario per aver cambiato idea su di me negli ultimi due anni, da " questo non dura più due mesi" a "chi se lo sarebbe aspettato".
Ma desidero ancora di più ringraziare tutti i collegiali del Don Mazza; Ringrazio Sebastian per il supporto didattico datomi per agevolare il mio percorso universitario, ma sopratutto per le tante bevute che ci siamo fatti, indimenticabile la notte di fuoco e follie a Venezia.
Ringrazio Vincenzo per la sua disponibilità, sempre e ovunque, e per il suo supporto, e non ti preoccupare che quest'anno te lo faccio vincere il torneo del vino, compare.
Ringrazio i mie nipotini Angelo ed Enrico aggiungendo anche Angjelo, ci siamo divertiti anche con voi, spero di avervela insegnata qualcosa. Bellissima l'esperienza della sceneggiata al mazzurro, mi sono divertito tanto, e sicuro abbiamo fatto divertire tutti gli altri, abbiamo ricevuto molti complementi. Grazie ad Angelo per essere venuto ad Eindhoven, spero che anche tu ti sia divertito, e ti verrò a trovare a Liegi.

Speciale il ringraziamento a tutto il terzo piano, specialmente a: Piero, miglior coppia difensiva mai vista al torneo del vino, cannavaro e nesta, bellissimo quando prendevamo gol e uno incolpava l'altro dell'accaduto.
Brune, in prima sessioneeeeeee.... vediamo se un giorno mi spiegarai
Bomber, immense le partite a calcetto, il torneo vinto insieme, e il fantacalcio, biabianyyyyyyy
Giane, vediamo se un giorno imparerai a non correre a cazzo di cane a calcetto.
Ma ringrazio anche Vitellino, Angelone, Damiano, Maiella, e tutti gli altri per avermi accettato in questo piano, e di aver condiviso la gioia della vittoria del torneo del vino, non segnando neanche un gol e sbagliando anche il rigore in finale.
Un grande ringraziamento va a tutti gli altri allievi del Mazza, Erennio, Franky, Giovanni, il Dottor Randazzooooo, Zanette, Riccardo, i fratelli Parolin, Cristina, Paola, Elena, Mauro e tutti gli altri....
Un grande ringraziamento va a Ida e Costanza, grandi amiche che mi hanno sempre consigliato e supportato, grazie per le belle serate passate insieme.
Ringrazio a Giulia, vincitrice del Mimprendo, mi sono divertito tanto con te, sei uno spasso. Dopo aver vinto mimprendo dovevamo andare al redentore con il gommone, un giorno lo faremo.
Un grande ringraziamento va ai miei compagni d'avventura erasmus, che festoni che abbiamo fatto tutti insieme, Silvia, Stefania, Gianluca, Chiara, Anna, Dani, Maxime, Ledi, Tim, Martin, seriamo di vederci presto. Speciale il ringraziamento va a Stephan e Abi, i miei coinquilini ad Eindhoven, con il vicino di casa che veniva a minacciarci con un coltello da caccia, o gli atri che ci mandavano la polizia, spettacolo, 65ers for ever.
Un ringraziamento particolare va ad ungrande amico, Alessandro, il destino ci ha fatto incontrare ad Eindhoven, e quante cose abbiamo fatto, quante cose abbiamo rubato ai poveri ingenui olandesi. Ancora aspettano i soldi della squadra di calcio. Grazie di tutto e in bocca al lupo per la tua laurea.
Ringrazio tutte quelle ragazze che hanno reso questi anni più interessanti.
Concludo qua ringraziando tutte le persone non menzionate in queste poche pagine (che sono un'infinità) e che hanno contribuito a rendere questi anni meravigliosi. Ci rivedremo in questa vita o nell'altra.
Grazie di tutto!