Digital Signal and Image Management

Alberto Filosa

29/9/2020

Indice

1 Classificazione dei Segnali

I segnali possono essere classificati in base al Dominio ed al Codominio. Si riportano i seguenti esempi:

1. Segnale Analogico, $\mathbb{D} : \mathbb{R} \to \mathbb{R}$;
2. Segnale Analogico a tempo discreto, $\mathbb{D} : \mathbb{R} \to \mathbb{K}$, con $K = \{..., t-1, t, t+1, ...\}$;
3. Segnale Continuo nelle ampiezze, $\mathbb{D} : \mathbb{R} \to \mathbb{K}$;
4. Segnale Digitale, $\mathbb{D} : \mathbb{K} \to \mathbb{K}$.

Per rappresentare digitalmente un segnale analogico sono necessari 3 fasi:

- Campionamento: si considerano solamente le parti di segnali tali per cui non si perdono troppe informazioni. Una alta frequenza di campionamento significa una buona riproduzione del segnale originale, ma si avrà un numero elevato di dati, mentre una bassa frequenza di campionamento produce il fenomeno chiamato aliasing;
- Quantizzazione: l'ADC campiona una onda analogica ad intervalli temporali uniformi ed assegna u valore digitale ad ogni campione. Il valore è ottenuto tramite la seguente formula:

$$\text{Digital Output Code} = \frac{\text{Analog Input}}{\text{Reference Input}} \times (2^N - 1)$$

;

- Codifica, limitata dalla memoria del dispositivo digitale e dalla sua velocità. Il file verrà compresso, processato o trasmesso

Per un efficace trattamento dei segnali è necessario minimizzare il quantitativo dei dati processati individuando solo quelli strettamente necessari, in modo da perdere meno informazioni possibili. La differenza tra il segnale analogico e la sua rappresentazione digitale è definita rumore di quantizzazione; il rumore diminuisce all'aumentare dei bit richiesti per la codifica del singolo campione.

Per *Ampiezza* si intende il valore assunto dal segnale e sarà la variabile dipendente y. Il tempo (o lo spazio) corrisponde alla variabile indipendente x, monodimensionale o a più dimensioni.

Le grandezze statistiche utilizzate sono:

Energy: $E_f = \int_{-\infty}^{+\infty} f^2(t)dt$

Power: $P_f = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} (|f(t)|)^2 dt$

Average: $\mu = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)dt = \frac{1}{T_1 - T_0} \int_{T_0}^{T_1} x(t)dt$

Se il segnale è una forma d'onda ripetuta, queste escursioni sono costanti e possono essere descritte da una grandezza chiamata Ampiezza *Picco-Picco*.

La periodicità di un segnale indica il tempo nel quale è definito il segnale che si ripete. La frequenza fondamentale è legata al periodo della relazione $f_0 = 1/T$. Nella realtà non esistono segnali puramente periodici, ma segnali quasi periodici caratterizzati da forme d'onda che si ripetono quasi uguali.

Il *Decibel* è un'unità di misura di tipo logaritmico che esprime il rapporto fra due livelli di potenza. La misura in decibel tra due grandezze fisiche dello stesso tipo è quindi una misura relativa, adimensionale e non lineare:

$$Bel = \log_{10} \frac{P_1}{P_2} \implies dB = 10Bel = 20 \log_{10} \frac{A_1}{A_2}$$

2 Analisi di Fourier

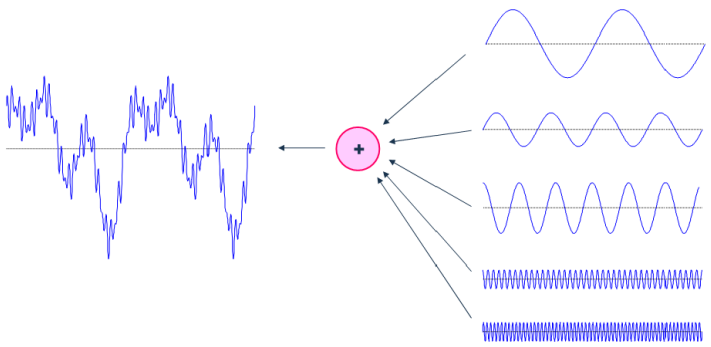
L'Analisi di Fourier ha lo scopo di decomporre il segnale in costituenti sinusoidali di frequenze differenti. In particolare, consente di osservare il segnale non più nel dominio tempo - spazio, ma nel dominio delle frequenze.

Ogni funzione periodica e a quadrato sommabile può essere espressa come somma di funzioni di seno e coseno:

$$y = A\sin(\varpi x + \phi) y = A\cos(\varpi x + \phi)$$

L'ampiezza indica quali sono i valori che la sinusoide può assumere, La pulsazione ($\varpi = 2\pi/T$) indica la frequenza della sinusoide, la fase indica quanto ritardo c'è nella sinusoide classica

Ad esempio, il segnale della immagine di sinistra è ottenuto sommando i segnali nella parte destra.



La serie di Furier scrive un segnale nella seguente forma:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(\frac{2\pi}{N} kx) + b_k \sin(\frac{2\pi}{N} kx)$$

con N periodo, $(1/N)$ definita come frequenza fondamentale f_0, k/n frequenza $f_k = kf_0$ e $\varpi = 2\pi f_k$ pulsazione. La formula diventa perciò:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(2\pi k f_0 x) + b_k \sin(2\pi k f_0 x)$$

Data la funzione $f(x)$ periodica, i coefficienti della serie sono univocamente determinati:

$$a_k = \frac{2}{N} \int_{-N/2}^{N/2} f(x) \cos(2\pi k f_0 x)$$
$$b_k = \frac{2}{N} \int_{-N/2}^{N/2} f(x) \sin(2\pi k f_0 x)$$

2.1 Trasformata

Ogni funzione continua $f(x)$, anche se non periodica, può essere espressa come integrale di sinusoidi complesse opportunamente pesate:

$$F(u) = \int_{-\infty}^{+\infty} f(x) e^{-j2\pi ux} dx f(x) = \int_{-\infty}^{+\infty} F(u) e^{j2\pi ux} dx$$

Inoltre, è possibile passare dalla trasformata di Fourier, definita come il dominio delle frequenze (o trasformato), alla anti-trasformata di Fourier, definita come dominio temporale. Questa trasformazione avviene senza perdita di informazione.

La trasformata di Fourier di una funzione continua ed integrabile è una funzione complessa nel dominio delle frequenze. In coordinate polari si ha:

$$F(u) = F[f(x)] = \Re(u) + j\Im(u) = |F(u)|e^{j\phi(u)}$$

Il modulo della trasformata $|F(u)|$ è definito come:

$$|F(u)| = \sqrt{\Re(u)^2 + \Im(u)^2}$$

mentre la fase $\phi(u)$:

$$\phi(u) = \tan^{-1} \frac{\Im(u)}{\Re(u)}$$

Per il caso bidimensionale l'equazione della trasformata assume un'altra forma:

$$F(u,v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x,y) \cdot e^{-i2\pi(ux+vy)} dxdy$$
$$f(u,v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(x,y) \cdot e^{i2\pi(ux+vy)} dxdy$$

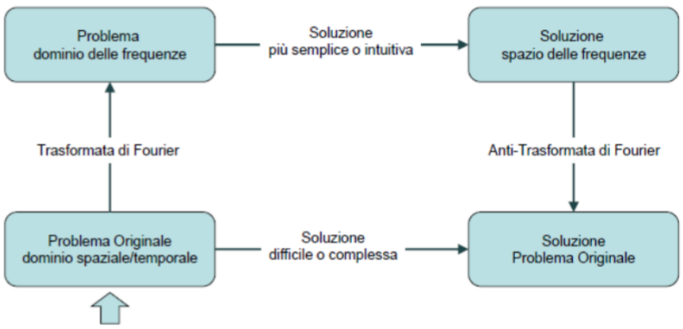
che rimane molto simile a parte per il numero di assi.

Operando con segnali digitali, che assumono valori discreti, l'integrale è sostituito dalla sommatoria:

$$F(u) = \frac{1}{N} \sum_{i=0}^{N-1} f(j) e^{-j2\pi u \frac{1}{N} i}$$

Si intende filtrare il suono di un'onda eliminando le frequenze sopra una certa soglia (ad esempio conservando solamente i bassi). Si effettua questa operazione nello spazio delle frequenze, molto più semplice in quanto si programma un filtro di una funzione che vale 0 sulle frequenze da eliminare e 1 quelle da conservare, e lo si moltiplica per la trasformata $F(u)$, effettuando l'anti-trasformata per poter fruire nuovamente del file.

Si può effettuare un filtraggio anche con le immagini, considerando solamente il modulo della trasformata bidimensionale del segnale. Più la rappresentazione del modulo è regolare, più l'immagine sarà ordinata. Per costruire un filtro basta considerare una circonferenza di raggio arbitrario per considerare solamente determinate frequenze. Considerando solamente le frequenze alte sono conservati i bordi dell'immagine, con frequenze basse le informazioni sul contenuto (si ottiene una sfocatura dell'immagine).



2.2 Convoluzioni

Una *Convoluzione* è l'operatore che descrive i filtraggi lineari nel dominio spaziale. In particolare, è l'applicazione di un filtraggio g ad una funzione f. Si considera una convoluzione nel dominio continuo tra due funzioni $f(x)$ e $g(x)$:

$$(f * g)(x) = \int_{s=-\infty}^{+\infty} g(x-s) f(s) ds$$

In particolare,

1. L'asse di rappresentazione di uno dei due segnali è invertita: $g(t) \to g(-t)$;
2. Il segnale invertito viene traslato tra $-\infty$ e $+\infty$;
3. Per ogni traslazione si calcola il prodotto del segnale traslato e quello non traslato;
4. Si calcola l'area del prodotto.

I filtri possono essere a media mobile, in inglese smoothing, nella quale i coefficienti sommano ad 1 ($\sum_i c_i = 1$), e derivativi, nella quale i coefficienti sommano a 0 ($\sum_i c_i = 0$)

Applicando la definizione di Trasformata di Fourier è possibile dimostrare il Teorema della Convoluzione:

La trasformata della convoluzione di due funzioni è il prodotto delle trasformate delle stesse:

$$G(u) = F[g(x)] = F[f(x) * h(x)] = F(u)H(u)$$

Per la corrispondenza fra dominio spaziale e dominio delle frequenze si hanno le seguenti relazioni:

Dominio Spaziale		Dominio delle Frequenze
$g(x) = f(x) * h(x)$	$\Leftrightarrow$	$G(u) = F(u)H(u)$
$g(x) = f(x)h(x)$	$\Leftrightarrow$	$G(u) = F(u) * H(u)$

3 Fondamenti di Immagini Digitali

Una *Immagine* è una funzione di intensità di luce a due dimensioni, $f(x, y)$, con coordinate spaziali rispetto alla luce in quel punto. Una *Immagine Digitale*, invece, è una rappresentazione di una immagine continua tramite un array bidimensionale di carattere discreto. Ogni elemento dell'array campionato è chiamato *Pixel*.

Il *Livello di Grigio* di una immagine è l'intensità relativa per ogni unità d'area, di solito compresa tra il valore più basso d'intensità (Nero = 0) e più intenso (Bianco = 255).

L'*Intensità* di una immagine è l'energia di luce, emessa da una unità d'area nell'immagine (dipende dal dispositivo), mentre la *Luminosità* di una immagine è l'apparenza soggettiva di una unità d'area (soggettiva e dipende dal contesto).

La *Luminance* è definita come potenza della luce pesata per una funzione spettrale chiamata efficienza luminosa. Essa mi

I pixel $f(x, y) = f_{yx}$ sono ordinati in modo naturale in una matrice, con x la colonna e y l'indice di riga:

$$\begin{bmatrix} f(0,0) & f(1,0) & ... & f(N-1,L-0) \\ f(0,1) & f(1,1) & ... & f(N-1,1) \\ \vdots & \vdots & & \vdots \\ f(0,L-1) & f(1,L-1) & ... & f(N-1,L-1) \end{bmatrix}$$

Il *Contrasto* di un punto di un'immagine è definito come la differenza relativa tra l'intensità del punto stesso e quella del suo vicino:

$$C = \frac{I_p - I_n}{I_n}$$

Il contrasto di una immagine intera, invece, è definito come la quantità di livello di grigio presente nella stessa. Per osservare il livello di grigio presente in una foto è necessario costruire un istogramma, che descrive la relativa proporzione di livello di grigio (assoluta, normalizzata e cumulata). è possibile manipolare il contrasto della intera immagine con lo *Stretching*, diminuendo il contrasto di un immagine, e *Shifting*, aumentando il valore di grigi.

3.1 Miglioramento Immagini

Il miglioramento delle immagini è il processo nella quale una immagine viene visualizzata meglio, ma che dipende dal tipo di problema da svolgere. Esso può essere effettuato sia nel dominio dello spazio, lavorando sulla immagine originale, che in quello delle frequenze, operando sulla trasformata di Fourier.

$$g(x, y) = T[f(x, y)]$$

Esistono diverse tecniche di miglioramento delle immagini:

- *Point Operations*, nella quale qualsiasi punto di una immagine dipende solamente dal livello di grigio in quel punto. Le operazioni più utilizzate sono:
  - Contrast Stretching, nella quale si aumenta il range dinamico di dei livelli di grigio nella immagine. I valori prima di una soglia m verranno compressi con una trasformazione $s = T(r)$;
  - Image Negatives, ovvero una trasformazione negativa della seguente espressione: $s = L - 1 - r$;
  - Compression of Dynamic Range, in quanto alcune volte è meglio comprimere i valori di grigio di una immagine. Di solito si utilizza una trasformazione logaritmica: $T(r) = c \log(1 + |r|)$, altre volte si utilizza la trasformazione $s = cr^{\gamma}$. Il fenomeno di correzione questa equazione è chiamata Gamma Correction;
  - Gray-Level Slicing, nella quale si focalizza l'attenzione su uno specifico range di valori di grigio. In questo caso è possibile evidenziare il range con un alto contrasto il range desiderato (Binarizzazione), oppure preservare la tonalità dello sfondo.
  - Bit Plane;
  - Histogram Operations, nella quale è possibile modificare il livello medio dell'immagine (*Sliding*), espandere/comprimere il range dinamico della scala di grigi (*Stretching/Shrinking*), o migliorare il contrasto di una immagine distribuendo equamente il livello di grigio di una foto.
  - Local Enhancement, nella quale scegliendo un intorno quadrato, si compie un istogramma locale e si applica l'equalizzatore a partire dal pixel centrale;
- Mask Operation, un filtraggio lineare di una immagine f che forma una sotto-immagine:

$$g(x, y) = \sum_{s=-a}^{a} \sum_{s=-b}^{b} w(s, t) f(x + s, y + t)$$

Lo Smoothing è una operazione di filtraggio nella quale si sfuma l'immagine riducendo la variazione dei pixel in scala di grigi.

4 Color Spaces

Un *Modello di Colore* è un modello matematico astratto che descrive come i colori possono essere rappresentati tramite numeri, tipicamente composto da 3 o 4 componenti. Lo *Spazio di Colore* è un assortimento di 3 dimensioni di colori nella quale ogni colore è rappresentato tramite un punti. Ne esistono di diversi tipi: il più importante è chiamato Device Derived, utilizzato per descrivere il livello del display del dispositivo, utilizzando il colore *RGB*.

I colori possono essere additivi, a partire dal livello di colore nero ed aggiungere i colori primari (rosso, verde e blu), oppure sottrattivo, a partire dal bianco e sottrarre i complementi dei colori primari (ciano, magenta e giallo).

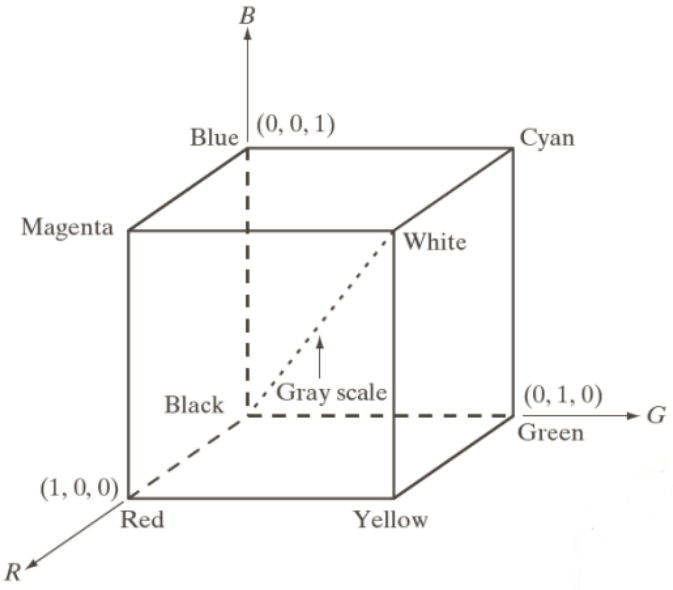
I colori in definitiva possono essere definiti come una combinazione dei colori primari:

$$\text{Color} = r\text{*R*} + g\text{*G*} + b\text{*B*} = \begin{bmatrix}\text{*R*} & \text{*G*} & \text{*B*}\end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

Il modello *RGB* è Device Oriented, ovvero la rappresentazione dei colori dipende dal tipo di dispositivo e riguarda sia l'acquisizione, ovvero il valore del colore dipende dalla sensibilità spettrale del sensore della camera, che l'esposizione, ovvero il colore RGB appare differente se visto da un altro dispositiva.

Molte volte è consigliato normalizzare i valori:

$$I = \frac{R+G+B}{3} \qquad r = \frac{R}{R+G+B} g = \frac{G}{R+G+B} \qquad b = \frac{}{}$$

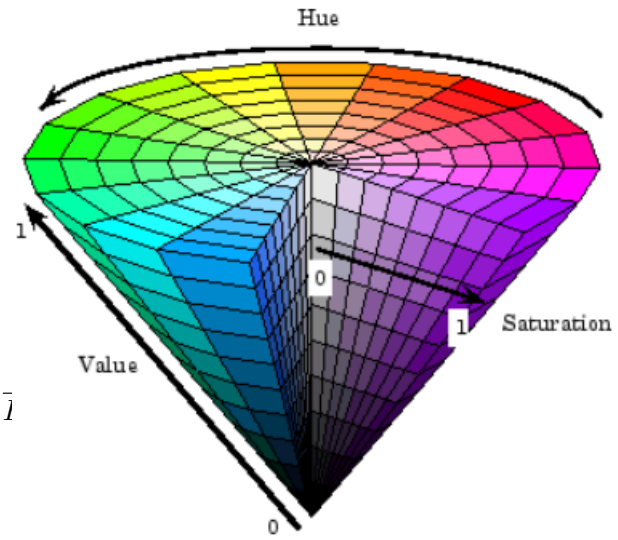


L'acquisizione delle immagini da parte di un dispositivo applica una Gamma correction per trasformare la luce da termini lineari a non linear, mentre i display applicano l'operazione inversa.

Un altro spazio di colori molto famoso è chiamato *Intuitive*, basato sulla descrizione dei colori familiari. Il modello di colore associato è ***H***ue ***S***aturation ***I***ntensity (**HSV**):

- *Tonalità*: indica la posizione del colore;
- *Saturazione*: indica le coordinate radiali della ruota dei colori
- *Intensità*: indica la quantità di luce.

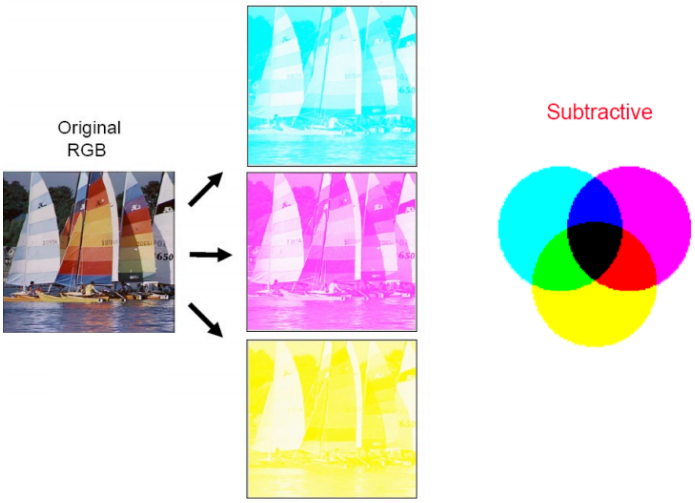
Questo spazio di colore è molto famoso in quanto più intuitivo e permette di compiere delle trasformazioni più efficienti rispetto dello spazio di colore precedente. Il problema principale è che quando una delle tre componenti è circa 0, le altre componenti diventano instabili.



Un altro modello molto utilizzato è il modello di colori *CMY*, spesso utilizzato per stampare fogli. Esso è il complemento del modello RGB: Al posto di aggiungere colori al nero, si sottraggono i colori dal bianco.

- *Ciano*, controlla la quantità di colore rosso nella stampante;
- *Magenta*, controlla la quantità di colore verde;
- *Giallo*, controlla la quantità di colore blu.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



5 Interest Point Detectors and Descriptors

Uno degli aspetti principali della Data Science è riconoscere oggetti presenti nelle immagini. In particolare, si chiama *Object Recognition* un classe di oggetti da dover identificare, mentre *Istance Recognition* una particolare istanza della classe dell'oggetto.

Un modo intuitivo per riconoscere un oggetto all'interno di una immagine è individuare tramite la rotazione, traslazione, illuminazione e qualità della immagine in inglese chiamato *Template Matching*. In particolare, è possibile utilizzare due approcci:

- ***S***um of ***S***quared ***D***ifferences (**SSD**), un metodo molto veloce, ma sensibile alla intensità generale della immagine;
- *Normalized Cross Correlation*, più lenta della precedente, ma non varia al variare della intensità di luce locale e al contrasto.

5.1 Local Descriptors Trends

I *Local Descriptor* sono dei descrittori che permetto di estrarre delle feature all'interno delle immagini a livello locale. Il più importante e famoso descrittore è lo ***S***cale ***I***nvariant ***F***eature ***T***ransform (**SIFT**), che permette di individuare oggetti anche a scale diverse all'interno della scena.

I descrittori locali sono *Keypoints-based* in quanto sono metodi efficienti per applicazioni in tempo reale. Sono in grado di comprimere informazioni in immagini di grande dimensione e riconoscere parti specifiche delle foto. Inoltre, gli algoritmi non devono compiere nuovi addestramenti (*traininig*) ogni volta che si applica la funzione.

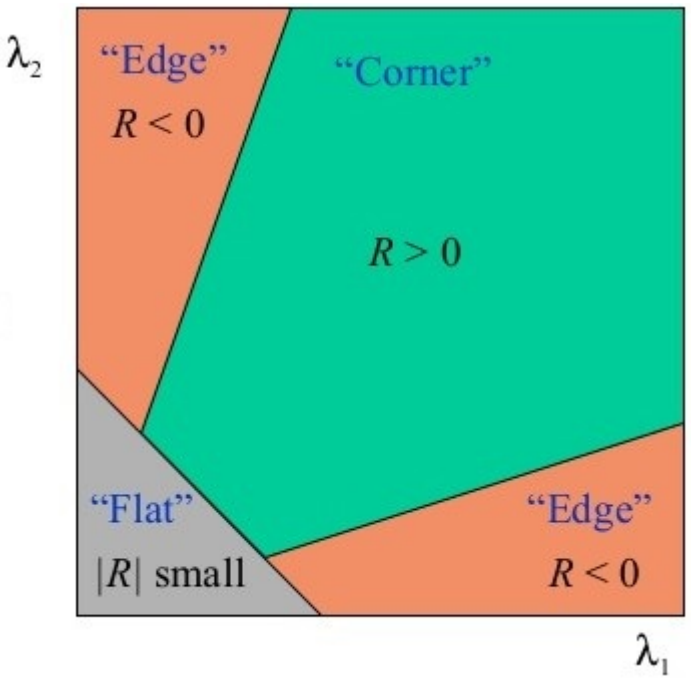
I punti più importanti all'interno di una immagine sono chiamati *Corner*, in quanto si verificano i maggiori cambiamenti in termini di luce e colore al loro esterno. La funzione che identifica questi punti è chiamata *Harris Corner Detector*, deinifita come la differenza tra l'immagine originale e la finestra in movimento tra le coordinate x, y:

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2$$

che grazie alla espansione di Taylor è possibile scriverla nella seguente formula matriciale:

$$M = \sum_{(x,y)\in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Gli autovalori della matrice sono utili per determinare il vero cambiamento del punto:



I passaggi fondamentali per la costruzione di un descrittore sono:

1. *Key-point Detection*: si costruisce lo Scale Space in modo tale da ottenere l'invarianza di scala (alto valore = alto blur, ma meno dettagli). L'immagine viene blurrata con dei filtri Gaussiani di intensità sempre maggiori, utilizzando diverse a scale e a diverse ottave. Queste immagini filtrate vengono sottratte una con l'altra per ottenere l'immagine differenza In questo modo si identificano i massimi e minimi dei punti. Esso è un candidato Keypoint se ha il valore più alto di tutto il suo vicinato. Si eliminano i candidati a basso contrasto, sotto il valore di una certa soglia, mantenendo solamente i valori con alto contrasto sia in termini verticali che orizzontali. Infine, si calcola l'orientazione dei punti per ottenere una invarianza di rotazioni, calcolando magnitudine ed orientazione dei punti. Verrà costruito un istogramma identificando il valore massima di orientazione;
2. *Key-point Description*: si costruisce una finestra 4×4 attorno al Keypoint selezionato, in ciascuna di queste celle viene calcolato l'istogramma per determinare l'orientazione. Si ottiene una descrizione dell'intorno del keypoint ottenendo 128 valori per descrivere l'intorno del punto. Il descrittore viene diviso per la sua norma $L2$ e viene calcolata l'invarianza per rotazione. Tutti i valori superiori a 0.2 vengono assegnati con il valore stesso e successivamente il vettore viene rinormalizzato;
3. Confronto dei Key-point simili (*Keypoint Matching*): i punti descrittori delle due immagini vengono confrontati utilizzando la distanza euclidea;
4. Punteggio di similarità basato sui punti di matching (*Score*): per confrontare la similarità tra due immagini è necessario mettere le immagini sullo stesso piano (*Omografia*). il metodo di confronto dei punti è chiamato ***RAN***dom ***SA***mple ***C***onsensus (**RANSAC**). Si seleziona la funzione che meglio approssima i punti, eliminando quelli più lontani perchè potenzialmente outlier. Il punteggio di score è definito come il rapporto delle distanze con il primo ed il secondo match identificato e si sommano tutti i match identificati. Il valore ha una tolleranza imposta dall'utente e se la supera allora le due immagini contengono lo stesso oggetto.

La metrica Metriche di valutazione più importante è chiamata *Intersection of Union*, definita come il rapporto tra l'area di intersezione e di unione di due immagini. Si inserisce una soglia al valore calcolato e per ciascun punto si identificherà un TP, TN, FP o FN.

$$IoU = \frac{\text{Area di Intersezione}}{\text{Area di Unione}}$$

6 Trainable Classifiers

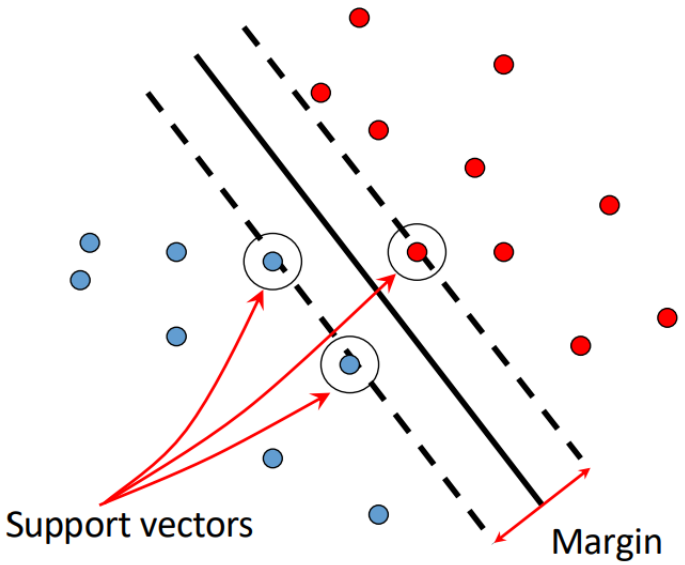
Il paradigma per riconoscere dei pattern all'interno di immagini è il seguente: a partire da un insieme di immagini di training si estraggono le principali feature di esse. Successivamente di classificano le immagini per allenare il classificatore che verrà poi utilizzato per successive immagini.

1-NN è un classificatore nella quale i descittori sono visti come un insieme di punti in uno spazio vettioare k-dimensionale. Successivmanete viene scelta una funzione di distanza (Euclidea, Mahnattan) per identificare la topopogia dello spazio

6.1 Support Vector Machine

Un altro classificatore molto famoso è chiamato ***S***upport ***V***ector ***M***achine (**SVM**), un modello che separa i valori di due (o più) classi diverse. Se i dati possono essere linearmente separabili, esistono più rette che separano le classi. In questo caso l'SVM individua il piano che massimizza il margine tra i valori delle classi.

$$wx + b = \sum_i \alpha_i y_i x_i x + b$$



Certe volte i punti non possono essere linearmente separabili. Per risolvere questo problema, si utilizza il *Kernel Trick*: se nello spazio di partenza i dati non sono linearmente separabili, lo possono essere in uno spazio vettoriale di dimensione superiore a quello di partenza:

$$\sum_{i=1}^{n} \alpha_i y_i \phi(x_i)\phi(x) + b = \sum_{i=1}^{n} \alpha_i y_i K(x_i x) + b$$

6.2 Neural Network

Le ***N***eural ***N***etwork (**NN**) sono dei modelli molto avanzati ispirati dai neuroni biologici che costituiscono il cervello animale.

Negli anni '60 Hubel e Wiesel hanno scoperto che le feature rispondono a pattern di attivazione in celle di basso livello e propagano l'attivazione a neuroni di una più alta gerarchia.

Un ***M***ulti-***L***ayer ***P***erceptron (*MLP**) consiste in un diverso numero neuroni artificiali che comunicano in modo unidirezionale, dalle variabili di input X all'attributo di classe. In generale, si calcola come la combinazione lineare tra le variabili di input meno la soglia (treshold):

$$y_j = f(\sum_{i=1}^{n} w_{ij} \times x_i - \theta_j) = f(z_j - \theta_j)$$

- Neuroni di Input, associato alle covariate;
- Neuroni Nascosti;
- Neuroni di Output, associato all'attributo di classe.

Ogni neurone di input è connesso in modo unidirezionale ai neuroni nascosti, propagando il segnale dal layer di input a quello nascosto. Quando tutti i neuroni del layer nascosto ricevono il segnale dai layer di input, il segnale è mandato a quello di output. Tuttavia, è possibile anche avere più di un layer nascosto: il primo layer manda tutti i segnali al secondo che si attiveranno. Alla fine, quest'ultimo manda dei segnali al neurone di output.

Il *Teorema di Approssimazione Universale* afferma che la rete Feed-Forward multistrato standard con un singolo layer nascosto, che contiene un numero finito di neuroni nascosti, è un approssimatore universale tra funzioni continue su sottoinsiemi compatti di R^n, sotto lievi ipotesi sulla funzione di attivazione.

Per determinare il minimo di una funzione è possibile utilizzare il *Gradient Descent*. Si sceglie la direzione d_k da seguire partendo da un punto x_k; successivamente, si massimizza (o minimizza) lungo la direzione il valore al fine di trovare un nuovo punto $x_{k+1} = x_k + \alpha_k d_k$, con k il numero di iterazione e α_k chiamata *Step Size*. I passaggi da seguire sono:

1. Scegliere un punto iniziale x_0;
2. Calcolare $\nabla f(x_k)$ alla k-esima iterazione;
3. Calcolare il vettore di ricerca $d_k = \pm \nabla f(x_k)$;
4. Calcolare il punto successivo $x_{k+1} = x_k \pm \alpha_k d_k$;
5. Utilizzare un metodo di risoluzione univariata per ottenere α_k;
6. Determinare la convergenza utilizzando una tolleranza $|f(x_{x+1}) - f(x_k)| < \varepsilon_1$, oppure $||\nabla f(x_{k+1})|| < \varepsilon_2$.

In questo modo è possibile individuare i parametri che minimizzano la Loss Function del modello. Si individuano i pesi della rete che minimizzano il valore atteso e quello previsto dal training:

$$L(v) = \sum_{j=1}^{n} [(y_j - f_w(x_j)]^2$$

Di solito si aggiunge un valore di penalizzazione sui pesi della funzione obiettivo:

$$L(v) = \sum_{j=1}^{n} [(y_j - f_w(x_j)]^2 + \frac{\lambda}{2} \sum_{j=1}^{n} w_j^2$$

6.3 Convolutional Neural Network

Le ***C***onvolutional ***N***eural ***N***etwork (**CNN**) sono delle Neural Network utilizzate quotidianamente nel modificare immagini aggiungendo dei filtri, ad esempio per la sfocatura di una immagine. Le CNN aiutano a ridurre i parametri focalizzandosi sulla connessione locale e

costruendo i Convolutional Layer. Non tutti i neuroni sono completamente collegati, ma lo sono in un sottoinsieme nel layer successivo. Inoltre, i pesi vengono anche condivisi lungo la posizione spaziale.

Le CNN sono estrattori di feature in termini *feed-forward*, ovvero da un input di osservazioni si produce l'output. Inoltre, vengono trainate in maniera supervisionata sfruttando filtri convoluzioni ed utilizzando la Back Propagation sugli errori. Le CNN presentano i seguenti layer:

- *Layer Convoluzionale*, dei filtri lineari, locali (porziona l'immagine comprimendola), invarianti per traslazione. Vengono usati molti filtri per ottenere una rappresentazione più ricca dei dati. Così facendo si produce una nuova mappa di attivazione tramite la banca di filtri:

$$Y_{ijq} = y_q + \sum_{u=0}^{H-1} \sum_{v=0}^{W-1} \sum_{k=1}^{K} x_{u+i,v+j,k} F_{u,v,k,q}$$

- *Funzione di Attivazione*: dopo la convoluzione, vengono applicate le funzioni di attivazione (gating) per ottenere la mappa in uscita. Ne esistono di diverso tipo: la più importante è la sigmoide:

$$y = \frac{1}{1 + e^{-x}}$$

- *Layer Pooling Spaziale*: per ridurre il numero di parametri dei modelli, è necessario ridurre il numero di parametri applicando la media (o il massimo) di una feature in un vicinato applicato canale per canale. In questo modo si riduce anche la invarianza per traslazione aumentandone la robustezza;

- *Local Response Normalization*, nella quale si applica una normalizzazione del contrasto. In questo modo è possibile migliorare l'invarianza e la sparsità delle reti. L'LRN può essere applicato all'interno di un canale, normalizzando singoli canali vicini, oppure tra i canali, normalizzando i gruppi dei canali.

6.3.1 Training

Per Training di una CNN si intende identificare i pesi w per minimizzare gli errori interni alla rete. In particolare, si utilizza una funzione di *Score* che mappa i pixel della immagine per ciascuna delle varie classi. Successivamente si calcola la funzione di perdita tra le etichette di ground truth e le etichette stimate tramite il metodo di *Stochastic Gradient Descent*. Il processo si basa su due processi:

1. Forward Propagation, il processo che calcola output dato dalla rete dato il suo input;
2. Backward Propagation, il processo che calcola il gradiente della Loss Function rispetto a tutti gli altri parametri della rete neurale.

Uno dei problemi principali delle CNN è, vista la presenza di numerosi parametri nella rete neurale, prevenire l'overfitting del modello. Per farlo è possibile penalizzare la magnitudine dei pesi, spegnere casualmente singoli neuroni (Dropout), oppure compiere il processo di Data Augmentation. In particolare, si aumenta il numero di osservazioni nel training set utilizzando degli esempi jitterati, un set di trasformazioni (diverso per ogni task) applicate in maniera casuale alle immagini mantenendo la label del dato originale. Infine si confrontano i modelli osservando il learning rate per individuare quello che ha la minor funzione di perdita.

6.4 Transfer Learning

Tutte le architetture CNN hanno milioni di parametri da trainare, in più è necessario avere un gigantesco dataset. Se non è possibile, si può compiere un pre-training dei dati con un grande dataset ed utilizzare la CNN come fine-tuning oppure come feature extraction per l'obiettivo di interesse.

Si suppone di voler addestrare un modello, dopodiché si prende la parte iniziale della rete eliminando la parte finale e si inserisce la porzione di interesse. Durante il training, i pesi dei layer più vicini all'input non vengono conteggiati, quindi si stimano meno i parametri.

Una importante applicazione delle CNN è lo *Speaker Recognition*, ovvero identificare la persona che sta parlando da una registrazione audio.

7 Content Based Image Retrieval

Il Content Based Image Retrieval è un processo nel quale data una immagine di query si vuole generare una lista di immagini ordinata, dalle più inerenti a quelle minori. In particolare, si vuole confrontare con una metrica di similarità la rappresentazione della query con tutte quelle nel dataset ed ordinarla in base alla similarità.

Problemi:

- Tipologia del dataset;
- Tipologia di query;
- Definizione di similarità tra immagini;
- Valutazione quantitativa delle performance;
- Come rendere efficiente la ricerca di una query.

Esistono due tipi di domini di immagini principali:

- Narrow Domain, ovvero bassa variabilità nell'aspetto tra le immagini
- Broad Domain, la variabilità dell'apparenza delle immagini è poco prevedibile.

Per quanto riguarda la tipologia di ricerche, è necessario individuare il soggetto nella immagine (Target Search). In particolare, bisogna capire se il soggetto è presente nella immagine, sia scalata che ruotata. Seconda possibilità è la category search, ovvero individuare se i soggetti appartengono alla stessa categoria (Category Search). Terza possibilità è la ricerca per associazione (Search by Association), data l'immagine di query, si vuole individuare i soggetti nelle foto con la maggior similarità. Per individuare cosa vuole l'utente è possibile far uscire una foto in base a delle keyword (Google Immagini) oppure in base ad una immagine far tirare fuori le immagini più rilevanti (Query by Example). Molte volte è possibile anche formulare una query con più di una immagine per estrarre le feature richieste. Altre volte è possibile raffinare la ricerca raffigurare l'identikit della immagine richiesta (Query by Sketch).

Range Search, data la misura di similarità, è possibile calcolare la distanza di similarità e si vuole individuare tutte le immagini in un range. Un'altra possibilità di ricerca è con il k-NN, che identifica le k immagini che hanno la miglior similarità data la query.

7.1 Performance Evaluation

Per valutare un sistema *CBIR* è necessario definire un insieme di immagini come dataset di test. Per ogni query si ottiene un giudizio di rilevanza (relevance judgement) e si comparano i risultati ottenuti rispetto ai documenti rilevanti per una determinata query. Per misurare le performance si utilizzano:

- *Recall*, definito come il rapporto tra i documenti rilevanti nella collezione;
- *Precision*, definito come il rapporto tra i documenti rilevanti predette ed il total dei documenti rilevanti.

Tipicamente, precision e recall sono combinati tra loro formando la metrica chiamata *F-Score*, calcolata come media armonica tra le due metriche:

$$F_1 = \frac{2rp}{r+p}$$

I vettori di feature (feature vector) vengono confrontati con una metrica di distanza (Es: Euclidea, Manhattan, Mahalanobis):

$$d(x,y) = \sqrt[r]{\sum_{k=1}^{n} |x_k - y_k|^r}$$

Per individuare se due immagini sono simili si confrontano tutti i feature vector calcolando la distanza tra di esse:

$$d(X,Y) \sum_{i=1}^{n} w_i d_i(x_i, y_i)$$

I problemi principali di questa metrica è che i valori potrebbero essere molto alti, di conseguenza le immagini sono molto diverse, è necessario scegliere la distanza migliore per ogni feature e i pesi devono essere scelti accuratamente. Per risolvere questi problemi è possibile normalizzare le feature in un range $[0,1]$ e calcolare indici come la *Mean* o *Gaussian Normalization*.

Le immagini devono avere le seguenti proprietà:

- Similarità Percettiva, la distanza della feature tra due immagini è largo solamente se le immagini non sono simili;
- Efficienza, velocemente computabili;
- Economia, le immagini devono avere una dimensione ridotta;
- Scalabilità, le performance del sistema non deve essere influenzata dalla grandezza del database;
- Robustezza, non hanno effetti sulle performance di retrieval cambiando l'immagine.

Per ottenere una migliore robustezza della immagine è possibile dividere in regioni più piccole l'immagine stessa. In questo modo è possibile estrarre delle feature creando un Bag of Word, ottenendo informazioni riguardo la presenza di testi in esse ($\delta = 1$ se coppie simili, 0 altrimenti).

$$l(x_1, x_2, \delta) = \delta l_P(d_D(x_1, x_2)) + (1-\delta) l_N(d_D(x_1, x_2)) \quad = \delta d_D(x$$

Per apprendere una miglior rappresentazione della similarità tra immagini è possibile utilizzare il *Network Siamese*. La rete neurale traina una funzione che mappa i pattern di input in uno spazio che approssima la distanza semantica nella spazio di input. La funzione di perdita utilizzata è chiamata *Triplet Loss*, che minimizza a distanza tra la query di ricerca ed i valori positivi e massimizza la distanza tra la query e il negativo.

8 GAN