METODI E TECNICHE PER IL MIDDLEWARE

Attività 2.1 - b

Sunto

Analisi combinata di segnale video e speech per l'identificazione di contesti all'interno di contenuti video.

Attività 2.1- Metodi e tecniche per il Middleware

Deliverable (Technical Report): Analisi combinata di segnale video e speech per l'identificazione di contesti all'interno di contenuti video.

Sommario

Introduzione

Negli ultimi anni sono stati sviluppate diverse soluzioni per la segmentazione dei video. I primi tentativi in particolare erano volti a determinare in maniera del tutto automatica, i confini delle riprese all'interno del video dove, i confini delle inquadrature sono stati definiti come confini fisici dove avvengono i cambi di telecamera. Negli anni, sono stati proposti diversi algoritmi [1] per il cambio di scena ed i migliori si sono dimostrati anche molto precisi.

Avendo questi algoritmi un'elevata precisione, il problema del riconoscimento del cambio di scena è considerato risolto.

In questo contesto, una osservazione fondamentale è legata a com'è formato il video. Infatti, questo può essere formato sia da un numero elevato di scatti, che da una sola ripresa. La probabilità che gli utenti cerchino il cambio di scena a seconda di scene semanticamente significative costituite da una serie di scatti è molto alta. Rispetto agli scatti, che sono ben definiti, è molto più difficile definire quando una scena è semanticamente significativa.

Classificazione dei metodi di segmentazione delle scene

La segmentazione delle scene può essere affrontata seguendo diversi approcci, tra i più importanti abbiamo:

1. **visual-based**: segmentazione basata sul visivo;
2. **audio-based**: segmentazione basata sull'audio;
3. **text-based**: segmentazione basata sul testo;
4. **audio-visual-based**: segmentazione basata sull'audiovisivo;
5. **visual-textual-based**: segmentazione basata sul testo visivo;
6. **audio-textual-based**: segmentazione basata sull'audio-testo;
7. **hybrid approaches**: segmentazione basata su approccio ibrido.

La definizione di come si caratterizza una scena è stato un dibattito portato avanti per diverso tempo.

A tal proposito, tra i vari studi effettuati, una scena è definita come una serie di inquadrature temporalmente contigue, caratterizzate da collegamenti sovrapposti che collegano inquadrature con contenuti visivi simili [2]. Ulteriore definizione importante è stata quella secondo la quale definisco una scena come un segmento contiguo di dati visivi con una coerenza a lungo termine di cromaticità, illuminazione e suono ambientale [3].

Le definizioni precedenti utilizzano la nozione di scena calcolabile, poiché tutte queste proprietà possono essere facilmente determinate utilizzando caratteristiche audio e video di basso livello.

Con il passar del tempo, da quando la semantica ha incominciato ad assumere importanza anche nella segmentazione del video, la scena è stata definita come una sequenza di inquadrature semanticamente correlate e temporalmente adiacenti che rappresentano un concetto o una storia di alto livello [4] [5]. Questa definizione presta particolare attenzione alle definizioni delle scene nella letteratura cinematografica.

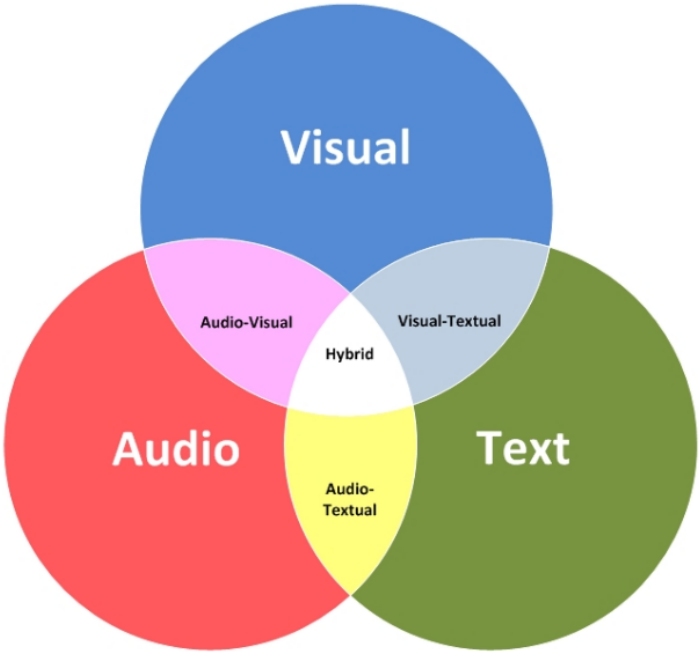

Figura 1: Categorie degli approcci per la segmentazione di una scena

Le informazioni basilari che sono disponibili quando ci si inizia ad approcciarsi ad un problema specifico di segmentazione, sono le caratteristiche che possono essere estratte e analizzate dal video.

La maggior parte degli approcci di segmentazione all'interno di una scena, hanno come fase iniziale quella del rilevamento della scena e quindi delle inquadrature. Nelle fasi successive del processo di rilevamento della scena, non vengono considerati tutti i fotogrammi del video, ma solo quelli chiave di ogni ripresa.

All'interno di ogni ripresa, il contenuto da un fotogramma all'altro non cambia significativamente, pertanto è sufficiente calcolare il confine della scena soltanto sulla base dei fotogrammi chiave. Per quanto concerne l'estrazione dei fotogrammi chiave, è possibile applicare diverse strategie: la più semplice riguarda la selezione del fotogramma centrale di ogni scatto.

Il processo fondamentale alla base di tutti gli algoritmi di scene detection è legato all'identificazione degli scatti che sono semanticamente coerenti e temporalmente vicini l'uno con l'altro.

Per far questo, è possibile identificare tre metodi di base:

1. **Rule-based methods**: metodi basati su regole;
2. **Graph-based methods**: metodi basati su grafici;
3. **Stochastic methods**: metodi stocastici.

Tali metodi sono utilizzati in combinazione con diverse caratteristiche di scene detection per la rilevazione delle scene.

La categorizzazione basata su caratteristiche di basso livello viene quindi ulteriormente affinata utilizzando le tre classi di metodi di segmentazione sopra citate e che vengono di seguito spiegate nel dettaglio:

Rule-based methods: metodi basati su regole

Un metodo per identificare le scene è quello di applicare **funzioni di somiglianza** al fine di raggruppare riprese simili all'interno di un intervallo di tempo predefinito.
Un ulteriore metodo è quello di non basarsi soltanto sulla vicinanza temporale, ma anche di considerare il modo in cui una scena è strutturata. Nella produzione cinematografica professionale ad esempio, i registi si affidano a determinate regole per la creazione delle scene. Queste regole vengono spesso definite come *regole di montaggio* o *grammatica cinematografica*.
Sono stati proposti diversi approcci di segmentazione delle scene video che tengono conto delle seguenti regole [6][7]:

- **Regola dei 180 gradi:** per posizionare le telecamere viene utilizzata una *linea immaginaria* (si veda la Figura 2). Tutte le telecamere sono posizionate su un lato della linea, catturando la scena soltanto da quel lato. In questa maniera viene preservato il contesto sullo sfondo della scena.
- **Regola del matching delle azioni:** la direzione del movimento dovrebbe essere la stessa in due scatti consecutivi che registrano il movimento continuo di un utente.
- **Regola del tempo del film:** il *ritmo di una scena* viene rappresentato dal numero di inquadrature, dalla regolarità dei suoni e dal movimento all'interno delle inquadrature. All'interno di una scena il ritmo non dovrebbe cambiare. Nella maggior parte dei casi un ritmo veloce indica la presenza di una scena d'azione [8].
- **Regola shot/reverse shot**: una scena può consistere di scatti alternati. Un esempio tipico è un dialogo tra due persone. La telecamera si muove tra i due personaggi mentre stanno parlando. Ma sono possibili anche inquadrature alternate tra persone e oggetti di interesse.
- **Regola di stabilizzazione/rottura:** quando si stabilisce una scena, in un'inquadratura panoramica vengono introdotti la posizione della scena, i personaggi e gli oggetti coinvolti e le loro relazioni spaziali. Successivamente, le inquadrature di scomposizione mostrano dei primi piani, che scendono più nel dettaglio.
- **Regola di establishment/rottura:** quando definisco una scena, il luogo della scena, tutti i personaggi e gli oggetti coinvolti e le loro relazioni spaziali, sono introdotti in un'inquadratura panoramica. Successivamente, gli scatti di scomposizione mostrano i primi piani che vengono spesso descritti utilizzando la regola di stabilizzazione/rottura.

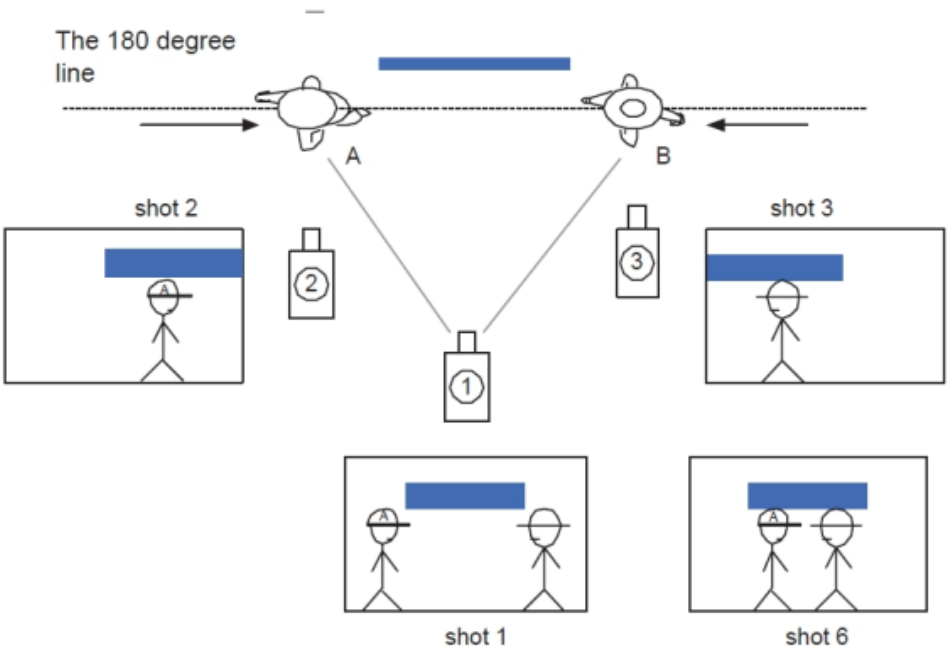

Figura 2: Esempio di segmentazione con regola dei 180 gradi

Graph-based methods: metodi basati su grafi

Fin dalla nascita della segmentazione delle scene video, sono stati introdotti approcci graph-based per la segmentazione delle scene. In questo caso il compito di rilevamento della scena viene trasformato in un problema di graph partitioning.

Gli algoritmi basati sui grafi hanno in comune il fatto che gli scatti vengono raggruppati in base alla somiglianza e nella maggior parte dei casi anche in base alla vicinanza temporale, ed in fine disposti in una rappresentazione grafica.

La Figura 3 mostra un esempio di questo approccio. I nodi rappresentano gli scatti o i cluster di scatti e i bordi indicano la somiglianza o la vicinanza temporale tra i nodi collegati. Applicando algoritmi di segmentazione dei grafi, quelli inizialmente costruiti vengono suddivisi in sottografi, ciascuno dei quali rappresenta una scena.

La segmentazione delle scene video con soluzioni graph-based funziona decisamente meglio per ambienti più piccoli, ristretti. In particolar modo per quei video che hanno scene che si ripetono più volte, come ad esempio i telegiornali o i talk show. Chiaramente, tale precisione risulta essere decisamente inferiore se il metodo viene applicato alle immagini in movimento.

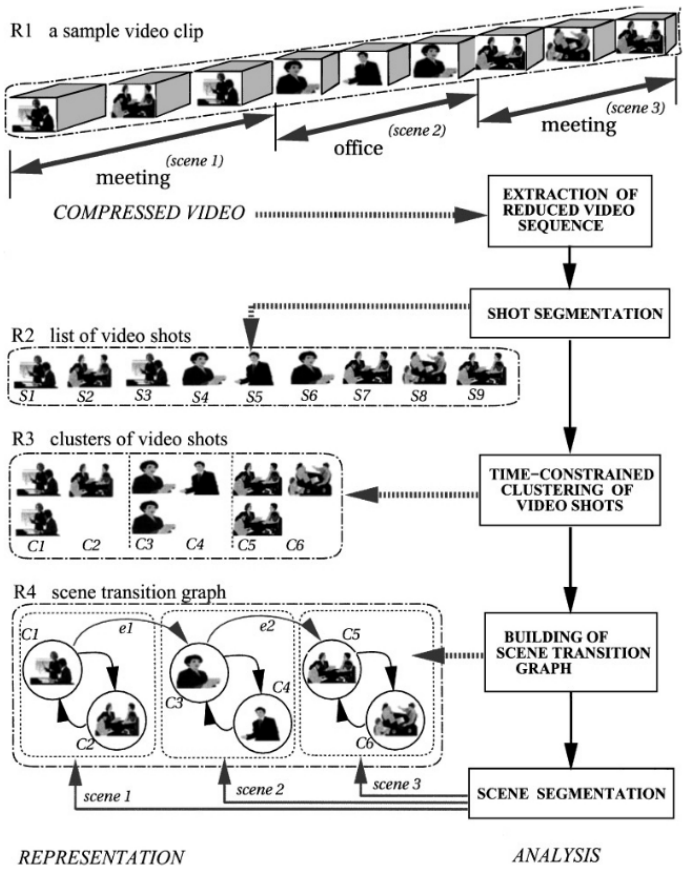

Figura 3: Metodo graph-based.

Stochastic-based methods: metodi stocastici

Gli algoritmi basati su metodi stocastici affrontano il problema della rilevazione dei confini di una scena utilizzando modelli stocastici. Una soluzione ottimale viene approssimata massimizzando la probabilità che i confini della scena stimata siano corretti. Grazie ad approcci basati sulla stocastica, è possibile ottenere un'elevata precisione.

Per determinare i modelli stocastici e per creare il set di formazione, sono necessari numerosi dati. Se il training set non viene accuratamente selezionato, gli algoritmi chiaramente non raggiungono risultati accurati. D'altra parte, anche se per la fase di training viene utilizzato un buon set di training, gli algoritmi falliscono comunque se applicati ai video, che hanno caratteristiche significativamente diverse rispetto ai video di training.

I metodi stocastici sono sempre limitati ad un dominio video ristretto, per il quale possono essere costruiti set di training più rappresentativi. Inoltre, se per la creazione dei vettori delle caratteristiche vengono utilizzate più caratteristiche di basso livello, è necessario valutare attentamente come queste caratteristiche possano essere meglio combinate.

Tecniche di Deep Learning per la segmentazione di scene

Negli ultimi anni, con l'avvento del **Deep Learning** (DL), la segmentazione delle scene ha subito numerose evoluzioni dal punto di vista algoritmico.

Infatti, i moderni sistemi intelligenti sono in grado di affrontare diverse situazioni senza l'intervento umano. È importante che i sistemi intelligenti conoscano il luogo o il contesto in cui si agisce, poiché li aiuta a capire cosa potrebbe essere successo in passato e cosa potrebbe accadere in futuro.

Un robot autonomo può essere considerato un esempio di tale sistema di intelligenza. Le strategie di adattamento dipendono dalle informazioni sull'ambiente nel passato, nel presente e nel futuro.

Infatti, il riconoscimento della scena invece di elencare gli oggetti presenti in questa, ha lo scopo di aiutare i calcolatori a comprendere gli ambienti che li circondano (pensiamo a quanto sia importante il riconoscimento della scena in uno scenario di Guida Autonoma).

Il riconoscimento delle scene è stato ampiamente utilizzato nelle applicazioni di interazione uomo-macchina, robotica intelligente, videosorveglianza intelligente e guida autonoma e viene inoltre considerato come un requisito fondamentale per altri compiti avanzati di visione artificiale, come ad esempio la detection di immagini e il rilevamento di oggetti.

L'obiettivo essenziale per il riconoscimento delle scene è quello di assegnare ad ogni immagine la propria *etichetta semantica.* Tali etichette, sono definite dall'uomo e possono comprendere diverse scene, ambienti e così via.

Nella Figura 4 sono in evidenza diversi esempi, ad esempio nella figura 4a, le immagini della classe Supermarket mostrano che le immagini delle scene possono contenere diversi oggetti. Nella figura 4b, le immagini della classe Coast (riga superiore) e Movie Theater (riga inferiore) sono utilizzate per illustrare le variazioni della disposizione spaziale. In fine nella figura 4c, le immagini di diverse categorie hanno ambiguità semantica ed appartengono rispettivamente a Chiesa, Chiostro, Biblioteca e Museo.

Figura 4: Esempi di immagini per il riconoscimento delle scene.

Algoritmo di Scene Detection

Prendendo come riferimento la manipolazione delle caratteristiche estratte dalle immagini, così come evince dalla Figura 5, gli algoritmi di scene detection possono essere raggruppati nelle seguenti sei categorie principali:

1. **Global Attribute Descriptors:** descrittori globali degli attributi;
2. **Patch Feature Learning:** codifica delle caratteristiche delle patch;
3. **Spatial Layouts Pattern Learning:** apprendimento del pattern di layout spaziale;
4. **Discriminative Region Detection:** rilevamento delle regioni discriminanti;
5. **Object Correlation Analysis:** analisi della correlazione degli oggetti;

6. **Hybrid Deep Models:** modelli ibridi profondi.

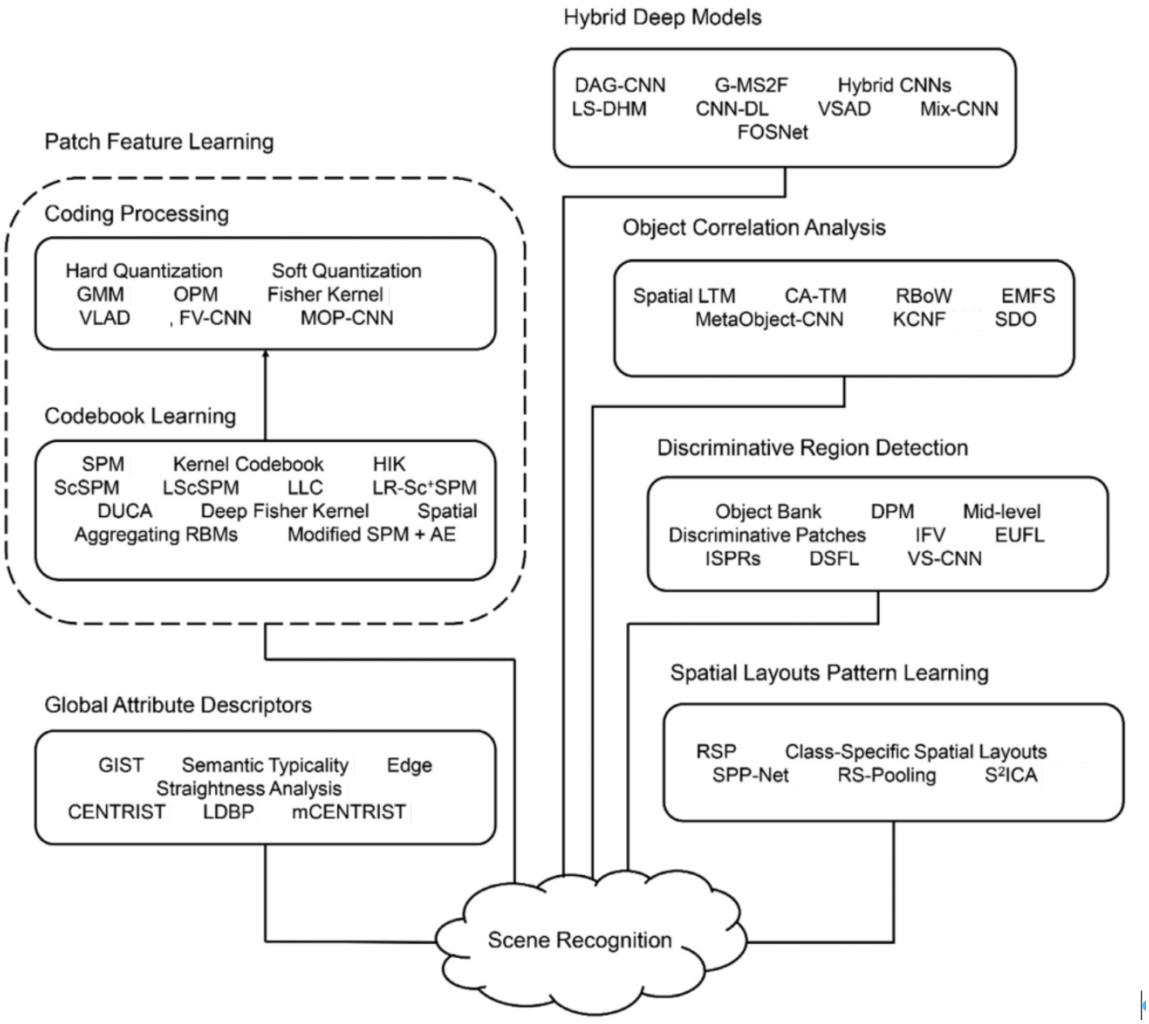

Figura 5: Tassonomia degli algoritmi di riconoscimento delle scene.

Nei primi anni 2000, la rappresentazione delle immagini di scena, si basava principalmente sui *Global Attribute Descriptors*, costruiti fondamentalmente da alcune proprietà visive di basso livello atte a modellare la percezione degli esseri umani. Tipici descrittori di attributi globali includono: GIST [9], *Semantic Typicality* [10], *Edge Straightness Analysis* [11], *CENsus TRansform hISTogram* (CENTRIST) [12], *Local Difference Binary Pattern* (LDBP) [13] and *multi-channel CENTRIST* (mCENTRIST) [14]. L'esecuzione di questi descrittori è chiaramente limitata da quelle che sono le costituzioni visive molto complesse delle immagini di scena.

Per migliorare le prestazioni di detection, molti ricercatori cominciarono a spostare l'attenzione su quello che è stato chiamato *Patch Feature Encoding*.

Nell'estrazione delle patch, sono stati utilizzati numerosi descrittori visivi, tra cui: *Local Binary Patterns* (LBP) [15], *Scale Invariant Feature Transform* (SIFT) [16], *Histogram of Oriented Gradients* (HOG) [17], *Speeded Up Robust Features* (SURF) [18] e *Oriented Texture Curves* (OTC) [19].

Inoltre, è stato introdotto il framework *Bag-of-Visual-Words* (BoVW) per integrare nella rappresentazione dell'immagine, un gran numero di descrittori visivi locali. Nel framework una componente di fondamentale importanza è quella che prende il nome di *Spatial Pyramid Matching* (SPM), ossia una componente standard per compensare le informazioni strutturali spaziali mancanti. In particolare, i descrittori visivi locali sono quantificati in parole visive finite, e quindi l'intera immagine della scena è descritta come la frequenza di occorrenza delle parole visive.

Il primo è possibile definirlo *codebook learning*, l'altro come *coding process*. La qualità del learned codebook ha un impatto notevole sulle performance della detection. Il kernel codebook [20] è proposto per sfruttare l'ambiguità tra le parole visive e per raccogliere maggiori benefici in uno spazio di caratteristiche con dimensione notevole.

L' histogram kernel [21] è introdotto con K-means al fine di ottenere un codebook learning più efficace ed una precisione di detection superiore. Si è inoltre scoperto che, gli algoritmi di clustering convenzionali nell'apprendimento dei codebook, sono molto sensibili agli outlayers e provocano grandi errori di quantizzazione.

Per superare questo problema, sono stati adottati nel dominio della scene detection, diversi algoritmi di apprendimento di tipo codebook learning basati sulla ricostruzione degli input. *Sparse coding* [22] ed i suoi derivati, sono stati sviluppati per imparare in maniera adattiva il codebook implicito e ridurre gli errori di quantizzazione.

Inoltre, alcune nuove architetture profonde (deep architecture) derivano da metodi convenzionali e sono proposte per codificare le caratteristiche delle patch locali, come ad esempio i fisher kernels [23] e diverse architetture profonde regolarizzate [24].

Le reti neurali profonde (Deep Neural Network) possono essere considerate come un metodo speciale di codifica delle caratteristiche dove il codebook corrisponde ai parametri di rete

appresi. Nel framework BoVW vengono state sfruttate le *Restricted Boltzmann Machines* (RBM) [25] e gli *Autoencoder* [26, 27].

Considerando il codebook, l'elaborazione della codifica aiuta a trasformare le parole visive locali nella rappresentazione dell'immagine. Uno dei metodi ampiamente utilizzati per l'elaborazione della codifica è la hard quantization o quantizzazione pesante o dura, in cui ogni parola visiva locale viene assegnata ad una parola visiva.

La *quantizzazione soft* [28,29] e il *Gaussian Mixture Model* (GMM) [30] vengono utilizzati per far fronte ai descrittori visivi locali in quanto assomigliano a più parole visive messe assieme.

Per integrare le informazioni spaziali nelle rappresentazioni di immagini, l'SPM è concepito per indicare le variazioni delle distribuzioni regionali delle parole visive. Analogamente, si combina lo spatial pooling multiscala con la codifica *Sparse coding Spatial Pyramid Matching* (ScSPM). A tal proposito, in combinazione con gli algoritmi di apprendimento del codebook, viene considerato in combinazione il modulo standard.

In aggiunta a quanto detto fino a questo punto, sono presenti anche alcuni lavori riguardanti la costruzione delle *piramidi spaziali*. La *piramide spaziale discriminante* [31] si propone di selezionare automaticamente i pesi di tutti i livelli della piramide per massimizzare il potere discriminativo. A tal proposito *Orientational Pyramid Matching* (OPM) [32] utilizza gli orientamenti 3D per indicizzare le patch nello spazio orientativo invece delle posizioni delle patch locali per formare la piramide. Recentemente, il kernel di Fisher [33, 34] e *Vector of Locally Aggregated Descriptors* (VLAD) [35] sono stati utilizzati per aggregare le caratteristiche convoluzionali delle patch in *Fisher Vector pooling* (FV-CNN) [36] e *Multi-scala orderless pooling* (MOP-CNNN) [37], mostrando delle ottime prestazioni.

Discriminative Region Detection

Il metodo del *Discriminative Region Detection* ha lo scopo di selezionare autonomamente alcune regioni che sono cruciali per il riconoscimento della scena. La Banca degli Oggetti [38] e i part-based models, modelli deformabili basati su parti [39] ricorrono ad algoritmi di rilevamento degli oggetti per ottenere le regioni discriminanti, mentre altri metodi come il clustering discriminante unsupervised [40], le *curve di entropia* [41] e la stima della densità [42], cercano di identificare le regioni discriminanti da un gran numero di patch di immagini.

Per sopprimere le caratteristiche con rumore o caratteristiche rumorose, l'apprendimento di importanti regioni di pooling spaziale o *important spatial pooling regions* (ISPR) [43] utilizza filtri parziali per mantenere la risposta su regioni ritenute fondamentali.

Per superare la varianza interna della classe e la somiglianza interclasse, caratteristica discriminante e condivisibile, l'apprendimento (DSFL) [44] viene sviluppato per far sì che le caratteristiche apprese dalla stessa categoria siano vicine l'una all'altra e le caratteristiche apprese da categorie diverse siano lontane l'una dall'altra.

Alimentando le CNN pre-addestrate con le immagini di scena d'importanza notevole, si sono dimostrate efficaci per il riconoscimento della scena le caratteristiche sensibili ottenute in profondità [45].

Object Correlation Analysis

L'*Object Correlation Analysis* cerca di modellare le relazioni tra la distribuzione di diversi oggetti e le categorie di scene. Le prime esplorazioni della correlazione degli oggetti si basano su modelli tematici in cui il riconoscimento degli oggetti è un prerequisito. Tipici modelli tematici o *topic model* sono il modello tematico latente spazialmente coerente o *spatially coherent latent topic model* [46], il modello tematico contestuale o *context aware topic* [47] e i modelli riconfigurabili o *reconfigurable models* [48].

Alla luce della distribuzione degli oggetti tra le diverse scene, alcuni modelli sono proposti per sfruttare gli schemi di co-occorrenza di ogni categoria come il modello di contesto nel collettore semantico o *context model in the semantic manifold* [49], *MetaObject-CNN* [50], *Kernel Co-occurrence Noise Filter* (KCNF) [51] e *Semantic Descriptor with Objectness* (SDO) [52].

Pipeline la Classificazione

Di seguito viene mostrata la pipeline per scene detection. Essa è formata da tre fasi fondamentali.

Il primo passo consiste nell'estrazione delle caratteristiche o *Feature Extraction*. Dopo aver acquisito le caratteristiche visive, la trasformazione delle caratteristiche viene adottata per catturare alcuni tratti della scena per formare la rappresentazione dell'immagine. Infine, la classificazione viene condotta utilizzando il classificatore.

Nei primi algoritmi di classificazione delle immagini, le caratteristiche visive sono rappresentate da caratteristiche molto rudimentali come bordi, angoli e vari descrittori visivi locali. Con l'avvento dei dataset di immagini su larga scala e il diffondersi delle *Deep Convolutional Neural Network* (DCNN), tali caratteristiche vengono gradualmente sostituite dalle caratteristiche profonde a causa della loro maggiore potenza espressiva.

La trasformazione delle caratteristiche o *Feature Transformation* è la tecnica più critica negli algoritmi di scene detection. Per la classificazione finale possono essere utilizzati alcuni metodi standard come SVM e softmax.

Data la rappresentazione dell'immagine, la performance di riconoscimento tra diversi classificatori a questo punto ha poca differenza.

Segue in Figura 6 la pipeline.

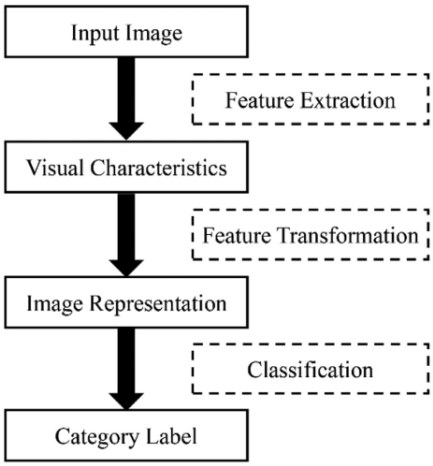

Figura 6: Pipeline per la Classificazione

Speech Transcription

La componente audio che correda un file multimediale risulta essere di enorme aiuto per l'identificazione delle scene da cui è composto. Basti pensare che molto spesso possono essere presenti dei fermo-immagine all'interno di un contenuto multimediale che però ospita un numero non unitario di scene. Questo può accadere quando a variare risulta essere il contesto del file, e dunque la scena stessa. Identicamente, tale avvenimento può accadere anche al contrario, ossia si manifesta un cambio repentino di caratteristiche dei fotogrammi, ma la scena rimane invariata.

Da questi semplici esempi si evince come la componente audio risulta essere fondamentale in quanto identifica una aggiuntiva dimensionalità per la definizione del contesto di una scena, della sua semantica e dunque della scena stessa.

L'estrazione automatica del contesto da un segnale audio non è un problema semplice da risolvere, in quanto si tratta di estrarre delle informazioni significative da segnali acustici provenienti da una sorgente vocale umana, dunque ambigua. Tale ambiguità è dovuta anzitutto alle molteplici modalità con cui anche un semplice termine può essere pronunciato, dal timbro mutabile della sorgente, rumori di sottofondo e soprattutto dal mezzo in cui tali informazioni sono convogliate, ossia il linguaggio naturale.

Il problema in questione viene solitamente affrontato suddividendolo in due sotto-problematiche, dia quali sono nati due ampi settori di studio, ossia lo Speech Transcription e il Natural Language Processing.

Il primo si occupa del riconoscimento dei termini ed intere frasi all'interno di un segnale audio appartenenti ad un linguaggio naturale, realizzando come output una trascrizione del parlato in una stringa, mentre il secondo campo studia quei modelli che siano in grado di estrarre il significato contenuto all'interno di un termine, frase o componimento. In entrambi i casi, tali discipline si occupano parallelamente anche di generare artificialmente un testo in linguaggio naturale avente una semantica precisa di partenza e la riproduzione audio dello stesso in modo tale che possa essere quanto più simile possibile al parlato di un essere umano.

Speech Recognition

Il parlato è il mezzo primario per eccellenza per la comunicazione tra persone [53], motivo per cui ogni file audio contenente un discorso risulta essere per definizione permeato di informazioni.

I primi tentativi di progettazione di sistemi per il riconoscimento automatico del parlato furono guidati dalla teoria dell'acustica-fonetica, descrivente i fonemi del parlato come unità minime e individua una loro descrizione al livello acustico su come vengono espressi. Essi individuavano un approccio basato su dizionari in cui alcuni termini venivano trascritti in base al risultato di una analisi parametrica della loro riproduzione audio. Il primo sistema che propose una ricerca ottimizzata della trascrizione più adeguata per i termini riprodotti, analizzati e segmentati fu

[54], che si avvantaggiò dell'algoritmo di *beam search* per raggiungere i migliori risultati del tempo per un sistema di quel tipo, ma solo alla fine degli anni '90 [55] possono individuarsi le prime metodiche funzionanti per una loro applicazione effettiva.

Soltanto in questo periodo, infatti, cominciò a diffondersi l'utilizzo dei modelli nascosti di Markov per gli scopi succitati. Tale metodica, che individua un processo doppiamente stocastico, è in grado di modellare la variabilità intrinseca del segnale del parlato, e le sue caratteristiche spettrali, così come la struttura stessa del linguaggio parlato all'interno di un framework statistico consistente. Il suo formalismo prevede una misura di probabilità sfruttante una catena di Markov per rappresentare la struttura linguistica del segnale, ed un insieme di distribuzioni di probabilità atte a ricoprire tutte le varianti acustiche dell'espressione del relativo sonoro.

A partire da un sufficiente insieme di espressioni etichettate, si adopera l'algoritmo di Baum-Welch [56] per ottenere i migliori parametri delineanti il modello di Speech Recognition, in maniera del tutto equivalente al processo di apprendimento dei più noti algoritmi di Machine Learning. Il modello risultante è in grado di indicare, attraverso il calcolo della verosimiglianza, se una espressione non conosciuta risulta essere la realizzazione audio effettiva di un termine specifico.

I modelli nascosti di Markov, infatti, tentano di identificare quelle caratteristiche di una sequenza probabilistica di osservazioni che potrebbe essere non statica, ma muta in dipendenza di una catena di Markov, successivamente estesa alla gestione di misture di densità ed unità ad una rete a stati finiti.

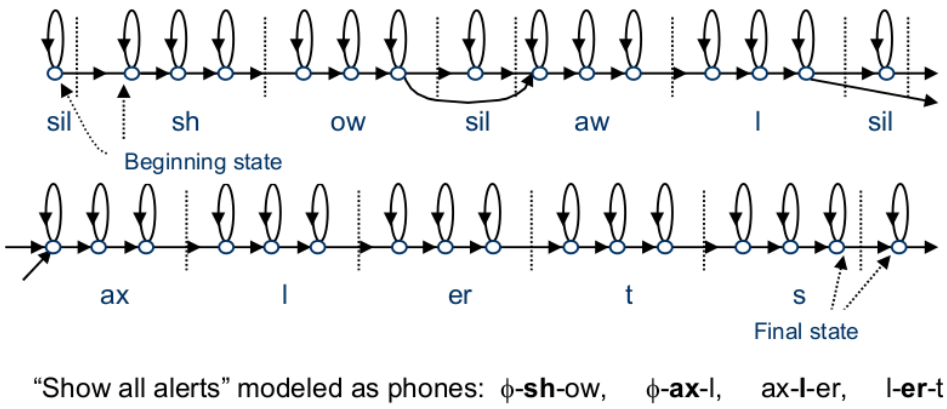

Figure 7: Rete a stati finiti per la espressione "show all alerts"

Parallelamente, con l'ottimizzazione dei modelli alla base delle celebri Artificial Neural Network, lo Speech Recogntion si confermò poter essere gestito in maniera ottimizzata esclusivamente tramite l'uso di modelli statistici, che assumono l'impossibilità di avere una forma prefissata delle espressioni linguistiche e, conseguentemente, possono essere riconosciute solo performando specifiche stime probabilistiche.

Deep Learning for Speech-to-text

L'utilizzo dei più recenti modelli di Deep Learning ha consentito lo sviluppo di svariati tool di Speech-to-text quasi sempre più performanti di quelli esemplificati precedentemente, compresi i modelli basati sui modelli nascosti di Markov estesi alle misture di Gaussiane.

Attraverso un modello di Deep Learning è possibile individuare un pattern latente per il riconoscimento di testo all'interno di un segnale audio più facilmente generalizzabile rispetto alle prime Reti Artificiali, risolvendo tale task come un semplice problema di multi-classificazione partendo da un ricco dataset su cui realizzare la fase di apprendimento profondo. Ciò permette al modello di modellare relazioni estremamente complesse e non lineari tra input e output [57] fondamentale nel campo dello Speech Recognition, se risulta ben gestita l'eventuale overfitting della rete sui dati di training.

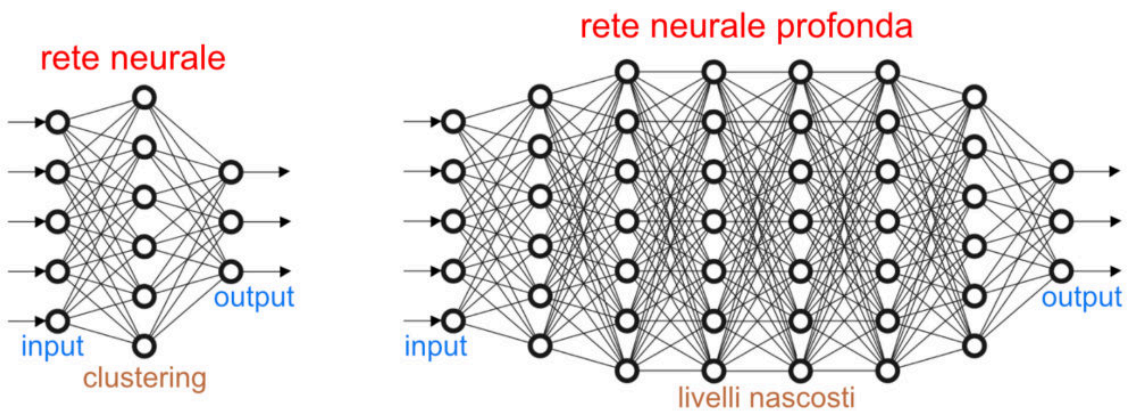

Figure 8: Differenza tra una ANN e una DNN

Altri esempi sono i Generative Pre-training Model, i quali si focalizzano sull'apprendere i pattern di fondo dei dati in input per la classificazione in testo piuttosto che l'apprendimento delle feature più discriminative per il succitato task di classificazione. In questo modo, attraverso una fase nota come fine-tuning, il modello che utilizza dei feature detector generici pre-addestrati si adatta, ad esempio, al riconoscimento del testo di una particolare lingua,

aggiornando i parametri intrinseci del modello stesso, o la Boltzmann Machine [58] e le Deep Belief Network [59].

In ogni caso, i modelli sono chiamati a risolvere il problema di multi-classificazione dei fonemi a partire da features, siano esse individuate da un'analisi del segnale audio in ingresso che ottenute da modelli pre-addestrati, solitamente valutabili sul dataset TIMIT[1]. Tale dataset fornisce un modo semplice e conveniente di testare nuovi approcci di speech recognition, di alcuni sono riportate le misure di accuratezza nella Tabella 1.

METHOD	ACCURACY
CD_HMM	27.3%
Augmented CRF	26.6%
Rand-RNN	26.1%
Bayesian Triphone GMM-HMM	25.6%
Monophone HTMS	24.8%
Heterogeneous Classifier	24.4%
Monophone Rand-DNN	23.4%
Monophone DBN-DNN	22.4%
Monophone DBN-DNN with MMI training	22.1%
Triphone GMM-HMM	21.7%

Table 1: Accuratezza nel riconoscimento fonetico su modelli addestrati con il dataset TMIT

Si è dimostrato, inoltre, come un pre-processing delle forme d'onda relativo all'audio su cui realizzare speech-to-text comporti un miglioramento delle performance sul task di TIMIT [60].

[1] https://catalog.ldc.upenn.edu/docs/LDC93S1/TIMIT.html

Bibliografia

[1] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. ACM Trans. Multimedia Comput. Commun. Appl., 3(1):3+, 2007.

[2] A. Hanjalic, R. L. Lagendijk, and J. Biemond. Automated high-level movie segmentation for advanced video-retrieval systems. IEEE Transactions on Circuits and Systems for Video Technology, 9(4):580–588, June 1999.

[3] H. Sundaram and S.-F. Chang. Computable scenes and structures in films. IEEE Transactions on Multimedia, 4(4):482–491, Dec. 2002.

[4] Y. Rui, T. S. Huang, and S. Mehrotra. Constructing table-of-content for videos. Multimedia Systems, 7(5):359–368, September 1999.

[5] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/Script: Alignment and Parsing of Video and Text Transcription. In D. Forsyth, P. Torr, and A. Zisserman, editors, Computer Vision ECCV 2008, volume 5305 of Lecture Notes in Computer Science, chapter 12, pages 158–171. Springer Berlin/Heidelberg, Berlin, Heidelberg, 2008.

[6] M. Ellouze, N. Boujemaa, and A. Alimi. Scene pathfinder: unsupervised clustering techniques for movie scenes extraction. Multimedia Tools and Applications, 47(2):325–346, Apr. 2010.

[7] B. T. Truong, S. Venkatesh, and C. Dorai. Scene extraction in motion pictures. IEEE Transactions on Circuits and Systems for Video Technology, 13(1):5–15, January 2003.

[8] B. Adams, C. Dorai, and S. Venkatesh. Toward automatic extraction of expressive elements from motion pictures: tempo. IEEE Transactions on Multimedia,4(4):472–481, December 2002.

[9] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelop, Int. J. Comput. Vis. 42 (3) (2001) 145–175.

[10] J. Vogel, B. Schiele, A semantic typicality measure for natural scene categorization, in: Proceeding of the Joint Pattern Recognition Symposium, 2004, pp. 195–203.

[11] A. Payne, S. Singh, Indoor vs outdoor scene classification in digital photographs, Pattern Recognit. 38 (2005) 1533–1545.

[12] J. Wu, J.M. Rehg, Centrist: a visual descriptor for scene categorization, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1489–1501.

[13] X. Meng, Z. Wang, L. Wu, Building global image features for scene recognition, Pattern Recognit. 45 (2012) 373–380.

[14] Y. Xiao, J. Wu, J. Yuan, mCENTRIST: a multi-channel feature generation mechanism for scene categorization, IEEE Trans. Image Process. 23 (2) (2014) 823–836.

[15] T. Ojala, M. Petikainen, D. Harwood, A comparative study of texture measures with classification based on feature distributions, Pattern Recognit. 29 (1996) 51–59.

[16] D.G. Lowe, Distinctive image features from scale-invariant key-points, Int. J. Comput. Vis. 60 (2) (2004) 91–110.

[17] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[18] H. Bay, T. Tuytelaars, L.V. Gool, SURF: speeded up robust features, in: Proceedings of the European Conference on Computer Vision, 2006, pp. 404–417.

[19] R. Margolin, L. Zelnik-Manor, A. Tal, OTC: a novel local descriptor for scene classification, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 377–391

[20] J.C. van Gemert, J. Geusebroek, C.J. Veenman, A.W.M. Smeulder, Kernel codebooks for scene categorization, in: Proceedings of the European Conference on Computer Vision, 2008, pp. 696–709.

[21] J. Wu, J.M. Rehg, Beyond the Euclidean distance: creating effective visual codebooks using the histogram intersection kernel, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 630–637.

[22] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1794–1801.

[23] V. Sydorov, M. Sakurada, C.H. Lampert, Deep Fisher kernels – end to end learning of the Fisher kernel GMM parameters, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1402–1409.

[24] Y. Yuan, L. Mou, X. Lu, Scene recognition by manifold regularized deep learning architecture, IEEE Trans. Neural Netw. Learn. Syst. 26 (10) (2015) 2222–2233.

[25] H. Goh, N. Thome, M. Cord, J.H. Lim, Learning deep hierarchical visual feature coding, IEEE Trans. Neural Netw. Learn. Syst. 25 (2014) 2212–2225.

[26] G. Xie, X. Zhang, C. Liu, Efficient feature coding based on auto-encoder network for image classification, in: Proceedings of the Asian Conference on Computer Vision, 2014, pp. 628–642.

[27] L. Xie, F. Lee, L. Liu, Z. Yin, Y. Yan, W. Wang, J. Zhao, Q. Chen, Improved spatial pyramid matching for scene recognition, Pattern Recognit. 82 (2018) 118–129.

[28] J.C. van Gemert, C.J. Veenman, A.W.M. Smeulder, J. Geusebroek, Visual word ambiguity, IEEE Trans. Pattern Anal. Mach. Intell. 32 (7) (2009) 1271–1283.

[29] Y. Boureau, F. Bach, Learning mid-level features for recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2559–2566.

[30] X. Zhou, X. Zhuang, H. Tang, M.H. Johnson, T.S. Huang, Novel Gaussianized vector representation for improved natural scene categorization, Pattern Recognit. Lett. 31 (8) (2010) 702–708.

[31] T. Harada, Y. Ushiku, Y. Yamashita, Y. Kuniyoshi, Discriminative spatial pyramid, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1617–1624.

[32] L. Xie, J. Wang, B. Guo, B. Zhang, Q. Tian, Orientational pyramid matching for recognizing indoor scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3734–3741.

[33] T. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, in: Proceedings of the Advances in Neural Information Processing Systems, 1998.

[34] J. Sanchez, F. Perronnin, T. Mensink, J.J. Verbeek, Image classification with the Fisher vector: theory and practice, Int. J. Comput. Vis. 105 (3) (2013) 222–245.

[45] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2012) 1704–1716.

[36] M. Cimpoi, S. Maji, A. Vedaldi, Deep filter banks for texture recognition and segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3828–3836.

[37] Y. Gong, L. Wang, R. Guo, Multi-scale orderless pooling of deep convolutional activation features, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 392–407.

[38] L. Li, H. Su, L. Fei-Fei, E.P. Xing, Object bank: a high-level image representation for scene classification and semantic feature sparsification, in: Proceedings of the Advances in Neural Information Processing Systems, 2010, pp. 1378–1386.

[39] M. Pandey, S. Lazebnik, Scene recognition and weakly supervised object localization with deformable part-based models, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 1307–1314.

[40] S. Singh, A. Gupta, A.A. Efros, Unsupervised discovery of midlevel discriminative patches, in: Proceedings of the European Conference on Computer Vision, 2012, pp. 73–86.

[41] M. Juneja, A. Vedaldi, C.V. Jawahar, A. Zisserman, Blocks that shout: distinctive parts for scene classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 923–930.

[42] Y. Yuan, J. Wan, Q. Wang, Congested scene classification via efficient unsupervised feature learning and density estimation, Pattern Recognit. 56 (2016) 159–169.

[43] D. Lin, C. Lu, R. Liao, J. Jia, Learning important spatial pooling regions for scene regions for scene classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3726–3733.

[44] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, X. Jiang, Learning discriminative and shareable features for scene classification, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 552–568.

[45] J. Shi, H. Zhu, S. Yu, W. Wu, H. Shi, Scene Categorization Model Using Deep Visually Sensitive Features 7 (2019) 45230–45239.

[46] L. Cao, L. Fei-Fei, Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes, in: Proceedings of the IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[47] Z. Niu, G. Hua, X. Gao, Context aware topic model for scene recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2743–2750.

[48] S.N. Parizi, J.G. Oberlin, P.F. Felzenszwalb, Reconfigurable models for scene recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2775–2782.

[49] X. Song, S. Jiang, L. Herranz, Multi-scale multi-feature context modeling for scene recognition in the semantic manifold, IEEE Trans. Image Process. 26 (6) (2017) 2721–2735.

[50] R. Wu, B. Wang, W. Wang, Y. Yu, Harvesting discriminative meta objects with deep CNN features for scene classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1287–1295.

[51] X. Song, S. Jiang, L. Herranz, Y. Kong, K. Zheng, Category co-occurrence modeling for large scale scene recognition, Pattern Recognit. 59 (2016) 98–111.

[52] X. Cheng, J. Lu, J. Feng, B. Yuan, J. Zhou, Scene recognition with objectness, Pattern Recognit. 74 (2018) 474–487.

[53] Juang, Biing-Hwang, and Lawrence R. Rabiner. Automatic speech recognition–a brief history of the technology development. Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara, 2005.

[54] B. Lowerre, The HARPY Speech Understanding System, Trends in Speech Recognition, W. Lea, Editor, Speech Science Publications, 1986, reprinted in Readings in Speech Recognition, A. Waibel and K. F. Lee, Editors, pp. 576-586, Morgan Kaufmann Publishers, 1990.

[55] M. Mohri, Finite-State Transducers in Language and Speech Processing, Computational Linguistics, Vol. 23, No. 2, pp. 269-312, 1997.

[56] L. E. Baum, An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes, Inequalities, Vol. 3, pp. 1-8, 1972.

[57] Geoffry Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath and Brian Kingsbury, Deep Neural Networks for Acoustic Modeling in Speech Recognition, IEEE Signal Processing Magazine, 2012.

[58] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput., vol. 14, pp. 1771–1800, 2002.

[59] G. E. Hinton, S. Osindero, and Y. Teh, A fast learning algorithm for deep belief nets, Neural Comput., vol. 18, no. 7, pp. 1527–1554, 2006.

[60] A. Mohamed, G. Dahl, and G. Hinton, Acoustic modeling using deep belief networks, IEEE Trans. Audio Speech Lang. Processing, vol. 20, no. 1, pp. 14–22, Jan. 2012.