Universitá degli Studi di Firenze

FACOLTÁ DI INGEGNERIA

Corso di Laurea in Ingegneria Informatica

Rappresentazione di video mediante Fisher Vector basati su contenuto e sentimento visuale

---

Candidato:
Andrea Garritano

Relatore:
Prof. Marco Bertini
Correlatore:
Dott. Andrea Ferracani

Febbraio, 2016

Indice

Capitolo 1

Introduzione

La crescente diffusione delle tecnologie informatiche e dei contenuti digitali ha permesso agli utenti di accedere ad un'enorme quantità di dati da qualsiasi punto del globo in modo estremamente semplice.
Da questo nascono però necessità di gestione della mole informativa e classificazione del media digitale.
Tale necessità ha reso necessario un interessante bisogno primario: la scelta di selezionare in modo efficiente la sovrabbondanza di informazioni.
Uno strumento molto potente che ci aiuta a combattere il fenomeno del "sovraccarico di informazioni" sono i sistemi di Information Retrieval.

Negli ultimi anni i sistemi di Information Retrieval hanno avuto una notevole espansione nel modo delle applicazioni software, grazie al continuo evolversi della tecnologia elettronica che ha permesso di ampliare le potenzialitá dei calcolatori.

Il mio lavoro si è incentrato sull'analisi dei contenuti e dei sentimenti visuali dei video utilizzando diversi modi di rappresentare quest'ultimi e sulla creazione di un sistema di Information Retrieval.

Capitolo 2

Strumenti

Per portare avanti il lavoro sono state usate tecnologie specifiche per i diversi compiti svolti: preparazione del dataset, sottocampionamento dei video, estrapolazione delle features e dei sentimenti visuali, analisi dei risultati.

2.1 Hardware

Per far funzionare al meglio gli algoritmi di raccomandazione si ha la necessità di avere un supporto hardware molto potente, in particolare una scheda grafica con prestazioni elevate.
Questo perché è necessario effettuare una mole molto grande di calcoli. Una CPU è costituita da diversi core ottimizzati per l'elaborazione seriale sequenziale mentre una GPU è dotata di una fitta architettura parallela costituita da migliaia di core di minori dimensioni e di maggiore efficienza progettati per la gestione simultanea di più operazioni.
Le reti neurali, utilizzate per svolgere la tesi, si prestano molto bene alla parallelizzazione dei task da svolgere.
L'hardware messo a disposizione è stato il seguente:

```
Processore:     Intel Xeon(R) CPU X5650  @ 2.67GHz
Scheda video:   nVidia GeForce GTX Titan
Scheda di rete: NetXtreme BCM5761 Gigabit Ethernet
Ram:            48GB ddr3
```

L'utilizzo di una GPU nVidia GTX Titan ha permesso di sfruttare i suoi 3072 CUDA Cores che hanno migliorato ancora di più le performance di calcolo.

CUDA è un'architettura hardware per l'elaborazione parallela creata da nVidia. Le applicazioni che supportano tale tecnologia sono capaci di eseguire calcolo parallelo sulle GPU delle schede video NVIDIA migliorando notevolmente le prestazioni di calcolo.

CUDA ha parecchi vantaggi rispetto alle tradizionali tecniche di computazione sulle GPU che usano le API grafiche.

- Il codice in running ottimizzato per Cuda può essere letto attraverso indirizzi arbitrari di memoria.
- CUDA espone una veloce condivisione di memoria che può essere condivisa fra i thread.
- Veloci downloads e riletture verso e dalla GPU.
- Supporto completo per divisioni intere e operazioni bit-a-bit.

L'utilizzo di Cuda ha permesso quindi un notevole incremento prestazionale per svolgere i calcoli necessari per analizzare i video.

2.2 Software

Oltre ad un buon supporto hardware, è stato necessario utilizzare un buon comparto software che ha permesso di progettare, sviluppare ed analizzare tutto ciò di cui avessi bisogno. Gli strumenti di sviluppo utilizzati sono basati su software open source, in quanto facilmente reperibili e adattabili al proprio scopo.

2.2.1 YoutubeDl

Youtube-Dl è un software che funziona da riga di comanda per scaricare video da YouTube.com e altre piattaforme web per la condivisione e visualizzazione in rete di video.
È un software open source e richiede l'interprete Python (2.6, 2.7, o 3.2+) per funzionare.
Youtube-dl è multipiattaforma, ma nel nostro caso è stato usato l'ambiente Unix. Il software è stato utilizzato per scaricare video, così da poter creare il dataset di dati necessari per svolgere il lavoro.

Il software è stato installato con il seguente comando da terminale:

```
sudo apt-get install youtube-dl
```

Per scaricare i video è stato necessario utilizzare questo comando dalla shell di comando:

```
youtube-dl -citk --max-quality FORMAT URL
```

Sostituendo URL con l'indirizzo della playlist di Youtube da scaricare.

2.2.2 FFmpeg

FFmpeg è uno dei principali framework multimediali open source in grado di decodificare, codificare, transcodificare, filtrare praticamente tutto ciò che gli esseri umani e le macchine hanno creato. Esso supporta una grande varietà di formati video ed è multipiattaforma.
Si basa su libavcodec, libreria per la codifica audio/video. FFmpeg è sviluppato su Linux, ma può essere compilato ed eseguito su qualunque dei principali sistemi operativi, incluso Windows.

FFmpeg è uno strumento da riga di comando per convertire un file video a un altro. Inoltre supporta la cattura e la codifica in tempo reale dalla scheda TV. Per installare l'ultima versione del software è stato necessario eseguire i seguenti comandi da terminale:

```
# Note: Super user privileges needed

# Remove any existing packages:
sudo apt - get remove ffmpeg x264 libav - tools libvpx - dev
    libx264 - dev


# Get the dependencies.
# (ubuntu Server or headless users):
sudo apt - get update

sudo apt - get -y install autoconf build - essential checkinstall
    git
libfaac - dev libgpac - dev libmp3lame - dev libopencore - amrnb -
    dev
libopencore - amrwb - dev librtmp - dev libtheora - dev libtool
libvorbis - dev pkg - config texi2html yasm zlib1g - dev

# x264 : H .264 video encoder
# The following commands will get the current
cd
git clone -- depth 1 git :// git . videolan . org / x264
cd x264
./ configure -- enable - static
make

# fdk - aac :
# AAC audio encoder .
cd
git clone -- depth 1 git :// github . com / mstorsjo / fdk - aac .
    git
cd fdk - aac
autoreconf - fiv
./ configure -- disable - shared
make
```

```
# libvpx :
# VP8 video encoder and decoder .
cd
git clone -- depth 1 http :// git . chromium . org / webm / libvpx
    . git
cd libvpx
./ configure
make

# FFmpeg : note , Ubuntu Server users
# should omit -- enable - x11grab . so :
cd
git clone -- depth 1 git :// source . ffmpeg . org / ffmpeg
cd ffmpeg
./ configure -- enable - gpl -- enable - libfaac -- enable - libfdk
     - aac --
enable - libmp3lame -- enable - libopencore - amrnb -- enable-
libopencore - amrwb -- enable - librtmp -- enable - libtheora --
    enable -
libvorbis -- enable - libvpx -- enable - x11grab -- enable -
    libx264 --
enable - nonfree -- enable - version3
make
```

Dopo l'installazione FFmpeg è stato utilizzato per sottocampionare i video scaricati.
Il sottocampionamento è stato fatto dapprima prendendo un frame ogni 3 secondi per ogni video, successivamente ogni secondo per evitare la perdita di contenuto visuale importante per la trattazione.

2.2.3 VLFeat

Il linguaggio di programmazione usato per progettare la gran parte del sistema di Information Retrieval è Matlab.
Matlab mi ha permesso di programmare in maniera veloce grazie al supporto delle librerie e di strumenti utili per il calcolo matematico.

Una libreria utilizzata per lo sviluppo della tesi è VLFeat.
VLFeat è una libreria open source che implementa algoritmi di computer vision specializzati nel riconoscimento delle immmagini ed estrazione di features locali. Sono implementati algoritmi come Fisher Vector, VLAD, k-means, Gmm e molti altri.
La libreria è scritta in linguaggio C e si interfacciano con Matlab.

Per installare l'ultima versione del software è stato necessario scaricare il pacchetto con l'ultima versione presente nella repository del sito uffiale di VLFeat (http://www.vlfeat.org/download.html), ed aggiungere al file startup.m presente nella directory di Matlab questa riga:

```
run('VLFEATROOT/toolbox/vl_setup');
```

Per controllare che l'installazione sia andata a buon fine è necessario scrivere nella shell di comando Matlab questo comando:

```
vl_version verbose
```

Nel caso di operazione eseguita con successo, VlFeat risponderà con un messaggio simile a questo:

```
VLFeat version 0.9.17
    Static config: X64, little_endian, GNU C 40201 LP64,
        POSIX_threads, SSE2, OpenMP
    4 CPU(s): GenuineIntel MMX SSE SSE2 SSE3 SSE41 SSE42
    OpenMP: max threads: 4 (library: 4)
    Debug: yes
    SIMD enabled: yes
```

2.2.4 MatConvNet

Un'altra libreria utilizzata è MatConvNet.
MatConvNet è una libreria che implementa algoritmi per le Reti neurali convoluzionali (CNNs) molto utili, anche'essi per applicazioni di computer vision.
Come VLFeat, è open source e si interfaccia con Matlab.

MatConvNet mette anche a disposizioni delle reti neurali già addestrate, come VGG-VeryDeep-16 ImageNet che mi ha permesso di estrapolare le features visuali delle immagini contenute nei video.

Il dataset della rete neurale VGG-VeryDeep-16 include immagini appartenenti a 1000 classi (concetti) addestrate su 1.3 milioni di immagini.
L'ultimo layer, infatti, da in output un vettore di pesi lungo 1000 che è stato preso ed analizzato.
Oltre all'ultimo layer, è stato usato anche il layer fc7 di 4096 elementi.

Per installare MatConvNet è necessario scaricare la libreria (http://www.vlfeat.org/matconvnet/), scompattarla e compilarla dalla shell di Matlab. Per sfruttare al massimo le potenzialità della GPU è stato necessario compilare la libreria specificando la directory del toolkit cuda, in caso non venga trovato automaticamente, e specificare di usare la GPU con questi comandi.

```
% install and compile MatConvNet (needed once)
untar('http://www.vlfeat.org/matconvnet/download/matconvnet-1.0-
    beta18.tar.gz') ;
cd matconvnet-1.0-beta18
run matlab/vl_compilenn('enableGpu', true, ...
               'cudaRoot', '/Developer/NVIDIA/CUDA-7.0', ...
               'cudaMethod', 'nvcc')

% setup MatConvNet
run  matlab/vl_setupnn
```

Per poter utilizzare la libreria è necessario inserire in ogni script Matlab l'esecuzione del setup di MatConvNet con il comando:

```
setup;
```

Per estrapolare le features di una singola immagine con la rete neurale preaddestrata VGG-VeryDeep-16 è stato usato il seguente script Matlab.

```
setup;

% load the pre-trained CNN
net = load('imagenet-vgg-f.mat') ;

% load an image
im = imread('peppers.png') ;

% preprocess image
im_ = single(im) ; % note: 0-255 range
im_ = imresize(im_, net.meta.normalization.imageSize(1:2)) ;
im_ = im_ - net.meta.normalization.averageImage ;

% run the CNN
res = vl_simplenn(net, im_) ;

% show the classification result
scores = squeeze(gather(res(end).x)) ;
[bestScore, best] = sort(scores, "descend") ;
```

Il codice prende come input l'immagine "peppers.png", la elabora con la rete neurale salvata in "imagenet-vgg-f.mat" e salva le features estrapolate nell'array "scores".
Successivamente, si ordina in modo decrescente gli score delle features e si possono stampare a video quali sono le features che rappresentano meglio il contenuto dell'immagine.

2.2.5 Caffe

Caffe è un framework per il deep learning sviluppato per essere uno strumento veloce, modulare e chiaro. Il progetto è nato da un'idea di Yangqing Jia e stato sviluppato dal Berkeley Vision and Learning Center (BVLC) e da una comunità di sviluppatori.

Usare Caffe per il deep learning ha notevoli vantaggi. L'architettura del framework è ben progettata e permette uno sviluppo veloce del codice. Passare da calcolo con la CPU alla GPU è semplice come settare un campo da vero a falso; così le reti neurali possono essere addestrate con una macchina che possiede una GPU molto potente e successivamente usate su un cluster di server.

Per installare Caffe è stato necessario avere una macchina che supportasse l'ambiente Cuda per poter processare le immagini in maniera veloce. Inoltre è stato necessario eseguire i seguenti comandi da terminale:

```
# Usage:
# 0. Set up here how many cores you want to use during the
    installation:
# By default Caffe will use all these cores.
NUMBER_OF_CORES=4
# 1. Execute this script, e.g. "bash compile_caffe_ubuntu_14.04.sh"
     (~30 to 60 minutes on a new Ubuntu).
# 2. Open a new shell (or run "source ~/.bash_profile"). You're
    done. You can try
#    running "import caffe" from the Python interpreter to test.

cd
sudo apt-get update
#sudo apt-get upgrade -y # If you are OK getting prompted
sudo DEBIAN_FRONTEND=noninteractive apt-get upgrade -y -q -o Dpkg::
    Options::="--force-confdef" -o Dpkg::Options::="--force-confold
    " # If you are OK with all defaults
sudo apt-get install -y libprotobuf-dev libleveldb-dev libsnappy-
    dev libopencv-dev libhdf5-serial-dev
sudo apt-get install -y --no-install-recommends libboost-all-dev
sudo apt-get install -y libatlas-base-dev
sudo apt-get install -y python-dev
sudo apt-get install -y python-pip git

# For Ubuntu 14.04
sudo apt-get install -y libgflags-dev libgoogle-glog-dev liblmdb-
    dev protobuf-compiler

# LMDB
# https://github.com/BVLC/caffe/issues/2729: Temporarily broken
    link to the LMDB repository #2729
#git clone https://gitorious.org/mdb/mdb.git
#cd mdb/libraries/liblmdb
#make && make install

git clone https://github.com/LMDB/lmdb.git
cd lmdb/libraries/liblmdb
```

```
sudo make
sudo make install

# More pre-requisites
sudo apt-get install -y cmake unzip doxygen
sudo apt-get install -y protobuf-compiler
sudo apt-get install -y libffi-dev python-dev build-essential
sudo pip install lmdb
sudo pip install numpy
sudo apt-get install -y python-numpy
sudo apt-get install -y gfortran # required by scipy
sudo pip install scipy # required by scikit-image
sudo apt-get install -y python-scipy # in case pip failed
sudo apt-get install -y python-nose
sudo pip install scikit-image # to fix https://github.com/BVLC/
    caffe/issues/50

# Get caffe (http://caffe.berkeleyvision.org/installation.html#
    compilation)
cd
mkdir caffe
cd caffe
wget https://github.com/BVLC/caffe/archive/master.zip
unzip -o master.zip
cd caffe-master

# Compile caffe
cp Makefile.config.example Makefile.config
sed -i '8s/.*/CPU_ONLY := 1/' Makefile.config # Line 8: CPU only
sudo apt-get install -y libopenblas-dev
sed -i '33s/.*/BLAS := open/' Makefile.config # Line 33: to use
    OpenBLAS
# Note that if one day the Makefile.config changes and these line
    numbers change, we're screwed
# Maybe it would be best to simply append those changes at the end
    of Makefile.config
echo "export OPENBLAS_NUM_THREADS=($NUMBER_OF_CORES)" >> ~/.
    bash_profile
```

2.2.6 DeepSentiBank

Infine è stato usato il tool DeepSentiBank.DeepSentiBank è un tool sviluppato dal Digital Video and Multimedia Lab della Columbia University in collaborazione con il Multimedia Analysis and Data Mining del Deutsche Forschungszentrum fur Kunstliche Intelligenz gGmbH (DFKI).

DeepSentiBank è una rete neurale pre-addestrata il cui output dell'ultimo layer completamente connesso è costituito da un softmax di 2089 input che generano una distribuzione di 2089 concetti visuali, ognuno dei quali formato da una coppia nome ed aggettivo.

Grazie all'ausilio del framework Caffe, DeepSentiBank mi ha permesso di estrapolare i sentimenti visuali delle immagini acquisite sottocampionando i video.

Per poter far funzionare DeepSentiBank è stato necessario ricompilare Caffe, inserendo il file "extract-nfeatures.cpp" presente nel pacchetto DeepSentiBank

nella cartella 'caffe/tools' di Caffe.
Una volta compilato è stato necessario linkare il file "extract-nfeatures.bin" compilato nella cartella "caffe/build/tools/".

Per estrarre le features è stato necessario creare uno script bash che richiamasse l'interprete python per ogni immagine da analizzare con il seguente comando:

```
python sentiBank.py test_image.jpg
```

L'output che il programma calcola è un file json contenente un array di 2089 concetti visuali, e un array contenente 4096 features corrispondente al layer fully connect 7 che è stato anch'esso utilizzato.

Per importare i dati in Matlab è stato usato il seguente codice:

```
%SentiBank
% id = numero video
% j = frame video
f = fopen(['/home/agarritano/Videos/' num2str(id) '/' num2str(j) '-
    features_prob.dat']);
features_senti = fread(f,[2089 1],'single');
fclose(f);
%End SentiBank
```

Capitolo 3

Realizzazione

Le fasi dello studio della rappresentazione di video mediante Fisher Vector basati su contenuto e sentimento visuale sono state molte. È stato necessario innanzitutto far propri i concetti di classification, clustering e recommendation. Successivamente è stato necessario capire come progettare del lavoro e creare il dataset da suddividere in test e training set.
Una volta create le basi per la vera e propria analisi, è stato fatta prima un'analisi con approccio Bag of Word, successivamente con strumenti più sofisticati quali Vlad e Fisher Vector.
Nel corso del lavoro è stata usata la Principal Component Analysis (PCA) per poter ridurre di dimensione i vettori che rappresentavano i video.

3.1 Information retrieval

L'Information retrieval è l'insieme delle tecniche utilizzate per ottenere una o più risorse rilevanti da una collezione di risorse informative quali documenti, pagine web e oggetti multimediali.

I sistemi di reperimento automatico delle informazioni sono molto utilizzati per combattere il fenomeno del "sovraccarico di informazioni".
I motori di ricerca quali Google, Bing e Yahoo sono le applicazioni di Information retrieval più usate. Numerose università e biblioteche usano i sistemi di Information retrieval per ricercare libri, giornali e documenti più pertinenti alle ricerche svolte dagli utenti.

Lo scopo dell'Information retrieval è quello di soffisfare il "bisogno informativo dell'utente", ovvero garantire a quest'ultimo, in seguito ad una sua ricerca, i documenti e le informazioni che rispondono alla sua richiesta.
In un sistema di Information retrieval sono di fondamentale importanza due concetti: query ed oggetto.

1. *Query:* Le query ("interrogazioni") sono parole-chiavi rappresentanti l'informazione richiesta. Vengono digitate dall'utente in un sistema Information retrieval e sono la concretizzazione del bisogno dell'utente.

2. *Oggetto:* Un oggetto è un item che possiede informazioni le quali potrebbero essere risposta dell'interrogazione dell'utente. Un video, per esempio, è un oggetto di dati.

A partire da una query svolta da un utente, il sistema di Information retrieval ha quindi il compito di fornire tutti i documenti rilevanti.
Molti sistemi di Information retrieval calcolano un punteggio numerico che rappresenta il grado di importanza e ordinano i risultati in ordine decrescente di punteggio. Un punteggio alto rappresenta un'entità che ha più probabilità di contenere il contenuto informativo richiesto dall'utente.

3.2 Preparazione Dataset

Per la preparazione del dataset sono stati scaricati video appartenenti a playlist di due macro categorie: Sport e Videogiochi. Ogni categoria è stata divisa in altre due sottocategorie: Calcio e Basket per la prima; Moba e Fps per la seconda.
Per il download dei video è stato utilizzato il software Youtube-Dl.
Ogni video scaricato è stato controllato per accertare che fosse nella categoria corretta ed evitare falsi positivi.

Con questo script bash i video sono stati rinominati con un identificativo numerico univoco.

```
# !/ bin / bash

k=1
for i in *.mp4; do
  new = $(printf "%d.mp4" "$k")
  mv -- "$i" "$new"
  let k=k+1
done
```

Successivamente i video sono stati sottocampionati con il software FFmpeg usando il seguente script bash:

```
#!/bin/bash

i = 1
nVideo = 457
while [ $i -lt nVideo ]
do
mkdir $i
ffmpeg -i $i.mp4 -vf fps=1 $i/%d.jpg
i=$[$i+1]
done
```

Figura 3.1: Esempi di keyframes

I video sono stati scelti interrogando Youtube utilizzando il filtro tipo "Playlist". Per quanto riguarda le categorie sportive sono state effettuate query con le seguenti parole chiave: "azioni salienti calcio", "azioni salienti basket". Mentre per la categoria videogiochi, sono state utilizzate le seguenti parole chiave: "league of legends gameplay" e "call of duty gameplay".

3.3 K-Fold

La K-fold cross-validation è una tecnica statistica che consiste nel suddividere il dataset totale in k parti uguali e, ad ogni passo, una delle k parti del dataset viene utilizzata come il dataset di test, mentre le k-1 parti rimanenti costituiscono il training dataset. Si evitando così problemi di campionamento asimmetrico del training dataset, garantendo una migliore distribuzione dei dati tra dataset di training e di test.

In altre parole, si suddivide il dataset in gruppi di egual numerosità, si esclude iterativamente un gruppo alla volta e lo si cerca di predire con i gruppi non esclusi. Ciò al fine di verificare la precisione del sistema di information retrieval.

Nel nostro caso è stata utilizzata la 10-fold cross-validation che è anche la più comunemente utlizzata, ma in generale K rimane in paramentro non fissato.
Quando K=N, con N numero di campioni del dataset, la k-fold cross-validation viene anche chiamata Leave-one-out cross-validation.

3.4 Approccio Bag of visual words

Il primo approccio utilizzato per la rappresentazione dei video è stato quello della Bag of Words, in particolare il modello Bag of Visual Words (BOVW).

Il Bag of Words è una rappresentazione semplificata degli oggetti usata in Information Retrieval.
Per rappresentare un'immagine con il modello Bag of Words, l'immagine può

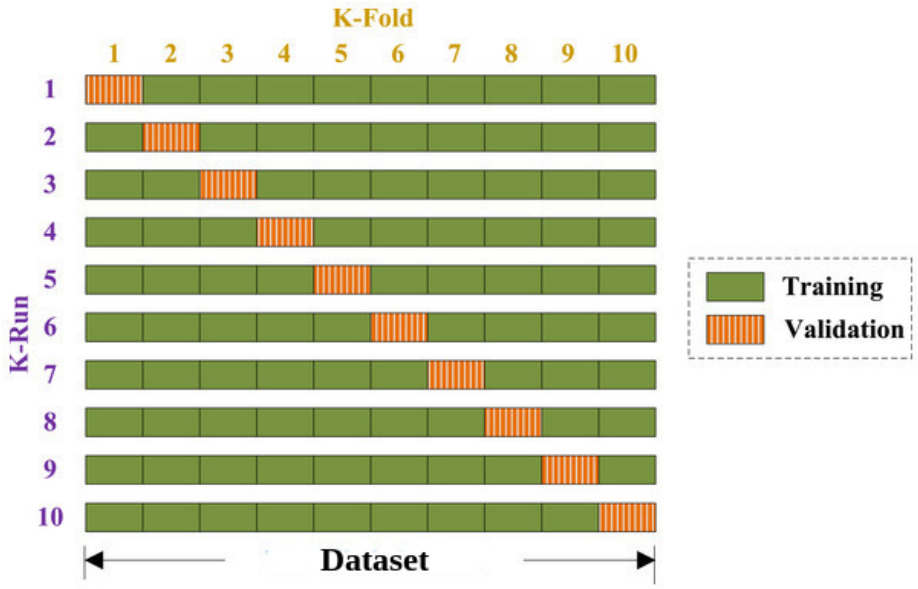

Figura 3.2: 10-Fold

essere trattata come un documento e le caratteristiche (features) rilevate in determinati punti dell'immagine si considerano "parole" visuali.

La Bag of Words è un vettore sparso del numero di occorrenze delle parole, che non è altro che un istogramma sparso sul vocabolario. In Computer Vision una Bag of visual words è un vettore di occorrenze del vocabolario di features locali dell'immagine.

Una volta estrapolate le features di ogni immagini, si normalizza l'istogramma e si procede cercando le immagini che abbiano caratteristiche simili.

Per quanto riguarda il lavoro svolto sulla rappresentazione dei video, è stato

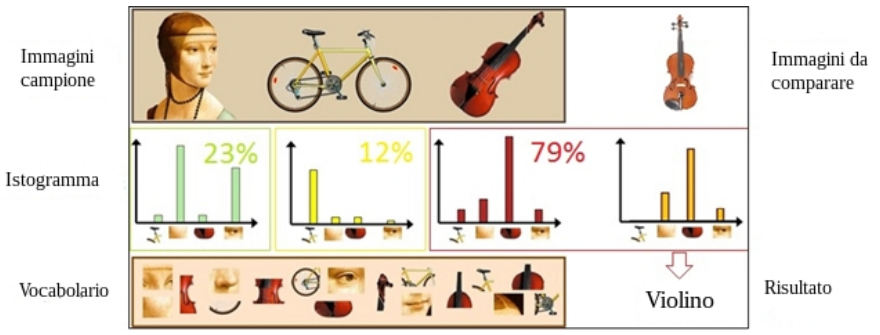

Figura 3.3: Bag of Visual Words

necessario l'utilizzo della rete neurale VGG-verydeep-16 per estrapolare le features di ogni frame sottocampionato per ogni video.
Successivamente ho normalizzato l'istogramma per ogni video con la norma L2 per fare in modo che la somma dei pesi dell'istogramma sia 1, così da poter confrontare istogrammi diversi.
Con questa normalizzazione la distanza coseno diventa il prodotto scalare tra vettori.

Per creare il vocabolario con i pesi di ogni video è stato implementato questo codice su Matlab:

```
setup ;

% load a pretrained model
net = load('data/imagenet-vgg-verydeep-16.mat') ;

resultMatrix = [] ;

d = dir(['/home/agarritano/Videos/', '*.mp4']);
nVideoTot = length(d(not([d.isdir]))) * 90/100;
nVideo = fix(nVideoTot) ;

for i = 1:nVideo
    d = dir([strcat('/home/agarritano/Videos/', num2str(i),'/'), '*
        .jpg']);
    nFrame = length(d(not([d.isdir])));

    weightVideo = single(zeros(1000,1)) ;
    for j = 1 : nFrame
        % obtain and preprocess an image
        fprintf('Bow:nVideo: %d/%d nFrame %d/%d\n',i,nVideo,j,
            nFrame)
        im = imread(strcat('/home/agarritano/Videos/', num2str(i)
            ,'/',num2str(j),'.jpg') );
        im_ = single(im) ; % note: 255 range
        im_ = imresize(im_, net.normalization.imageSize(1:2)) ;
        im_ = im_ - net.normalization.averageImage ;

        % run the CNN
        res = vl_simplenn(net, im_) ;

        % show the classification result
        scores = squeeze(gather(res(end).x)) ;
        weightVideo = weightVideo + scores ;
    end
    weightVideo = weightVideo *1/norm(weightVideo);
    resultMatrix = [resultMatrix weightVideo] ;
end
save('bowResultMatrixBpca.mat', 'resultMatrix')
```

Lo script non fa altro che caricare la rete neurale, calcolare le features per ogni immagine sottocampionata e creare i vettori di rappresentazione dell'immagine. Per ultima cosa, i risultati vengono salvati per essere riutilizzati in futuro.
Nel codice vengono estratte le features visuali prendendole dall'ultimo layer del-

la rete neurale.

Una volta creato un istogramma per ogni video, è stato possibile vedere la similarità tra video con questo codice Matlab che mostra i primi 10 video del dataset più simili ad uno dei video di test, ordinati dal più simile al meno simile.

```
max = 10;
% istogrammi dei video
load('bowResultMatrixBpca.mat');
resultMatrix = normc(resultMatrix);

% istogramma del video da testare
load('weightVideoTest.mat');

disp('Computing all distances...');
distance_type = 'KL2';
distance = vl_alldist2(resultMatrix, weightVideo, distance_type) ;
[y,idx] = sort(distance, 'descend');

disp(idx(1:10));
```

Nello script viene caricata la matrice che rappresenta gli istogrammi dei video, viene caricato l'istogramma del video su cui fare il test e calcolata la distanza KL2 tra ogni colonna della matrice "resultMatrix" e il vettore "weightVideo". La distanza KL2 non è altro che il prodotto scalare tra vettori.

3.5 Approccio Fisher Vector

Un approccio più sofisticato rispetto al semplice BoW utilizzato per la rappresentazione dei video è stato il Fisher Vector.
Il Fisher Vector è uno strumento più preciso rispetto al BoW perchè prende in considerazione le statistiche del primo e, facoltativamente, del secondo ordine.
Il FV tiene quindi conto rispettivamente della media e della varianza rispetto dei dati. Al contrario, contando il numero di occorrenze di concetti visuali, il BoW codifica solo le statistiche di ordine 0 della distribuzione dei descrittori.

Il vettore che contiene le statistiche del primo e del secondo ordine sono il gradiente delle probabilità dell'insieme dei descrittori rispetto ai parametri della distribuzione, diviso il reciproco della radice quadrata del Fisher Vector.
Queste informazioni ci danno la direzione nella quale la distribuzione appresa deve essere modificata per rappresentare meglio i dati.
In altri termini, il Fisher Vector descrive come il set di descrittori cambia rispetto alla distribuzione media dei descrittori.

Per poter creare la rappresentazione mediante Fisher Vector è stato necessario calcolare il Gaussian mixture model (GMM) su tutti i frame del training set.
Il Gaussian mixture model calcola una collezione di K distribuzioni Gaussiane sui frame del dataset e in output restituisce le medie, le matrici di covarianza e

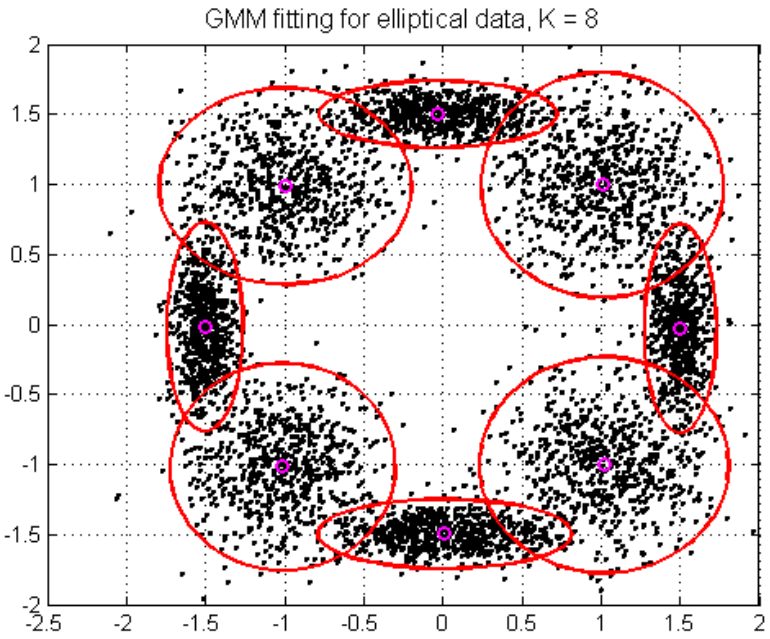

Figura 3.4: Esempio di GMM con K=8

la distribuzione di probabilità delle distribuzioni.

È stato utilizzato il seguente codice Matlab per fare ciò:

```
num_centers = 32;

tic;
disp('GMM computation for Fisher encoding...');
[means, covariances, priors] = vl_gmm(resultMatrix, num_centers);
save('fisherGMM.mat', 'means', 'covariances', 'priors');
toc;
```

La variabile "resultMatrix" rappresenta la matrice contenente tutti i frame del training set.

Successivamente è stato creato il vero e proprio Fisher Vector per ogni video del training set con questo codice Matlab:

```
enc = cell(nVideo,1);
```

```
for id = 1:nVideo
  resultMatrix = [] ;
  d = dir([strcat('/home/agarritano/Videos/', num2str(id),'/'), '*.
      jpg']);
  nFrame = length(d(not([d.isdir])));

  for j = 1 : nFrame
    fprintf('Fisher:nVideo: %d/%d nFrame %d/%d\n',id,nVideo,j,
        nFrame)

    f = fopen(['/home/agarritano/Videos/' num2str(id) '/' num2str(j
        ) '-features_prob.dat']);
    features_senti = fread(f,[2089 1],'single');
    fclose(f);

    features_senti=features_senti/norm(features_senti,2);

    resultMatrix = [resultMatrix features_senti] ;
  end

  %PCA code

  enc{id} = vl_fisher(resultMatrix, means, covariances, priors, '
      Improved');
end
save('encResultMatrixBpca.mat', 'enc')
```

Una volta creato il Fisher Vector per ogni video, come per il BoW, è stato possibile vedere la similarità calcolando la distanza KL2 tra i Fisher Vector.
Il Fisher Vector che è stato creato ha una dimensione pari al numero di centri della GMM per il numero delle features, moltiplicato per 2.

3.6 Principal Component Analysis

La dimensione del Fisher Vector risultata dalla computazione è molto grande, quindi è stato necessario ridurre le dimensioni del vettore.
Una tecnica utilizzata per fare ciò è la Principal Component Analysis (PCA).

Principal component analysis è una procedura che permette di analizzare un set di dati che potrebbero contenere variabili correlate e trasformalo in un set di variabili incorrelate chiamate componenti principali.

Questa procedura si basa sulla computazione di una nuova base della matrice contentenente i dati che possa rappresentare in modo più significativo quest'ultimi.
La nuova base viene scelta calcolando la matrice di correlazione tra tutte le coppie di valori della matrice originale. Vengono quindi prese in considerazione solo le componenti che hanno una varianza maggiore o in altre parole, le più importanti.

Il numero di componenti principali è minore o uguale alla cardinalità del dataset originale. Questo ci permette di ridurre notevolmente la dimensione dei dati

calcolati, mantentendo però le informazioni più salienti.

La PCA è stata utilizzata sia in fase di training dei dati, sia nella fase successiva al calcolo del Fisher Vector.

Questo è il codice Matlab utilizzato:

```
% pca_means.m
% descrs::          descriptors: 1 column per descriptor of row
    dimension
% dimensionality::  required dimensionality of the reduced
    projection. If
%                   between 0. and 1. it is considered as variance,
     and then
%                   it computes the dimensionality required to
    explain it.
% proj::            projection components
% descrs_mean::     descriptors' means

% Make sure data is zero mean
descrs_mean = mean(descrs, 2) ;
x = bsxfun(@minus, descrs, descrs_mean) ;
% Compute covariance matrix
X = x*x' / size(x, 2) ;
% Perform eigendecomposition of X
[V, D] = eig(X) ;
d = diag(D) ;
if exist('whitening', 'var') && (strcmp(whitening,'whitening'))
  V = diag(1./sqrt(d+1e-18)) * V;
end
[d, perm] = sort(d, 'descend') ;
% compute dimensionality that explains the required variance - code
     from
% Dimensionality Reduction Toolbox
if dimensionality < 1
    dimensionality = find(cumsum(D ./ sum(D)) >= dimensionality, 1,
         'first');
end
% select #dimensionality components associated with larger
    eigenvalues
m = min(dimensionality, size(descrs,1)) ;
V = V(:, perm) ;
proj = V(:, 1:m)' ;

% project_with_pca.m
% descrs:: original descriptors
% descrs_means:: original descriptor means computed using pca_means
% proj:: projection components computed using pca_means
descrs = proj * bsxfun(@minus, descrs, descrs_means) ;

% PCA
pca_percent = 1/16;
proj_size = round(size(resultMatrix,1) * pca_percent);
[ Dproj, Dmean ] = pca_means(resultMatrix, proj_size);
resultMatrix = project_with_pca(resultMatrix, Dmean, Dproj);
```

In questo caso la PCA è stata fatta per ridurre di 16 volte il dataset presente in "resultMatrix", ma sono stati portati avanti anche esperimenti con diversi fattori di riduzione.

3.7 Mean Average Precision

La valutazione di un sistema di information retrieval è il processo che descrive quanto il sistema è in grado di reperire le informazioni necessarie ad un utente.

In generale, le metriche considerano una collezione di oggetti in cui cercare e una query di ricerca.
Tutte le più comuni tecniche di valutazioni si basano sul fatto che ogni documento è noto per essere pertinente o non pertinente per una determinata query.

Praticamente tutte le tecniche di valutazione dei sistemi IR sono progettate per dare maggiore o minor peso ad un documento che viene reperito con un grado di pertinenza più o meno alto.

La tecnica di valutazione dei risultati ottenuti è stata la Mean Average Precision (MAP).

Per definire la Mean Average Precision è necessario definire la Precision.
La Precision misura la quantità di documenti pertinenti che soddisfano la necessità di informazione di un utente.

$$\text{Precision} = \frac{|\{\text{documenti pertinenti}\} \cap \{\text{documenti reperiti}\}|}{|\{\text{documenti reperiti}\}|}$$

Nel nostro caso, trattandosi di classificazione binaria, la Precision è analoga al numero di predizioni positive fratto il totale delle predizioni.
È necessario sottolineare che il significato di "precisione" nel campo dell'Information Retrieval differisce dalla definizione di accuratezza e precisione presente in altre branche della scienza e della statistica.

La Precision è una metrica che definisce un singolo valore rispetto l'intera lista di documenti reperiti dal sistema.
Per sistemi di Information Retrieval che classificano i documenti reperiti ordinandoli dal più pertinente al meno pertinente, come nel nostro caso, è sicuramente opportuno considerare anche l'ordine con cui i documenti sono presentati.

L'Average Precision tiene conto proprio di questo fattore. La formula per calcolare l'Average Precision è la seguente:

$$\text{AveragePrecision} = \frac{\sum_{k=1}^{n} (\text{Precision(k)} \times \text{rel}(k))}{\text{numero di documenti pertinenti}}$$

dove rel(k) è la funzione caratteristica che assume il valore 1 se il documento alla posizione K è un documento rilevante, il valore 0 se il documento non è

rilevante.
Da notare che la media è su tutti i documenti pertinenti e i documenti pertinenti non reperiti assumono una precisione uguale a 0.

La Mean Average Precision (MAP) per un gruppo di query è la media delle Average Precision per ogni query.

$$\text{MeanAveragePrecision} = \frac{\sum_{q=1}^{Q} \text{AveragePrecision(q)}}{Q}$$

dove Q è il numero di query.
Per calcolare la MAP è stato scelto di analizzare i primi 40 risultati del sistema IR.

Capitolo 4

Conclusioni

L'odiena rete internet permette da qualsiasi parte del mondo di accedere ad unae banca dati contenente un'infinità di dati multimediali di qualsiasi tipologia.
Lo scopo di questa ricerca si è posta l'obiettivo di riuscire a capire come poter soddisfare l'esigenza di informazione di un utente. In particolare sono stati analizzati alcuni dei più comuni modi di rappresentare i contenuti multimediali visivi e valutarne la precisione delle previsioni del sistema di Information Retrieval.

Durante lo studio sono stati utilizzati diversi approcci, ognuno dei quali è stato necessario per la comprensione dei risultati delle reti neurali messe a disposizione.
Una volta estrapolati i risultati dalle reti neurale, sono stati codificati e analizzate le predizioni del sistema IR.
È stato necessario combinare i risultati dei vari approcci e tecniche di codifica utilizzate per avere un quadro completo della situazione.
Analizzando i risultati si è potuto capire quali fossero le migliori combinazioni che rappresentassero nel miglior modo i video.

Gli esperimenti mostrano quale siano le precisioni del sistema IR utilizzando la metrica MAP.
Per prima cosa sono stati analizzati i risultati delle due reti neurali, analizzandone il layer Prob e Fc7. Sono state inoltre combinate.
È stata fatta una distinzione per quanto riguarda il numero di centri da calcolare attraverso il Gaussian Mixture Model.

Gli esperimenti condotti sull'intero dataset mostrano come cambiare il numero di centri non influisca a cambiare il valore della precisione. Anche l'utilizzo di layer diversi o di reti neurali diverse non cambia il valore della precisione.

I test però hanno riscontrato una netta separazione tra le varie categorie di dati.

Le diverse precisioni potrebbero essere causate da una minore omogeneità dei video nelle categorie sportive rispetto alle categorie Moba e Fps.
Mentre le prime due categorie sono caratterizzate da informazioni costantemen-

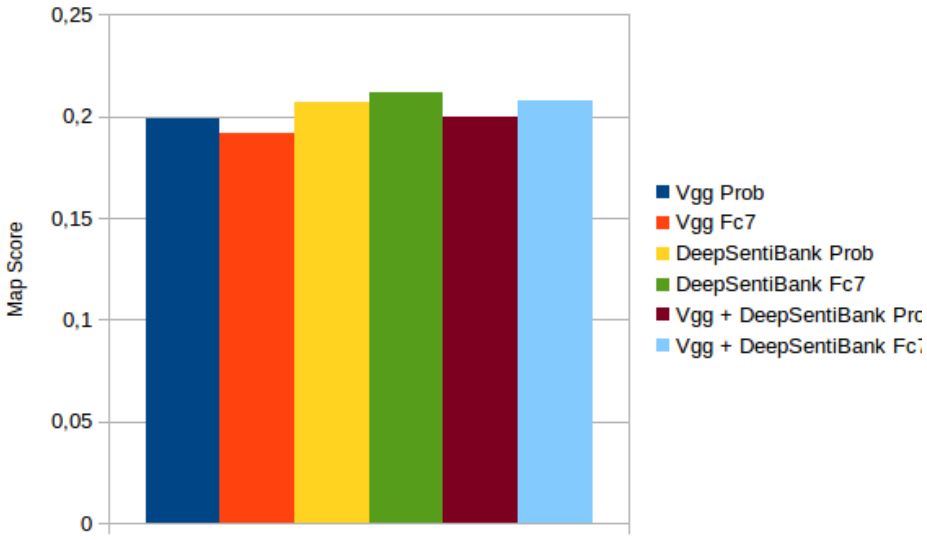

Figura 4.1: Grafico con numero centri GMM=32 Intero Dataset

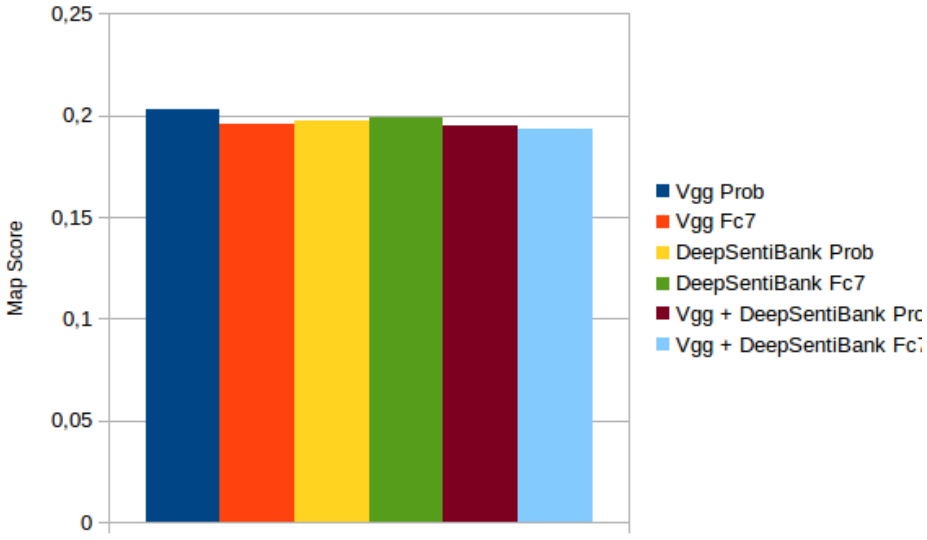

Figura 4.2: Grafico con numero centri GMM=16 Intero Dataset

Vgg Prob	0,199
Vgg Fc7	0,192
DeepSentiBank Prob	0,207
DeepSentiBank Fc7	0,212
Vgg + DeepSentiBank Prob	0,200
Vgg + DeepSentiBank Fc7	0,208

Tabella 4.1: Esperimenti con numero centri GMM=32 Intero Dataset

Vgg Prob	0,203
Vgg Fc7	0,196
DeepSentiBank Prob	0,197
DeepSentiBank Fc7	0,199
Vgg + DeepSentiBank Prob	0,195
Vgg + DeepSentiBank Fc7	0,193

Tabella 4.2: Esperimenti con numero centri GMM=16 Intero Dataset

te visibili, come l'Head-up display caratteristico dei videogiochi, le rimanenti non hanno elementi così fortemente caratterizzanti durante tutta la durata dei video.
Un'ulteriore disomogeneità è probabilmente causata dalla tipologia di inquadratura presente nei video che riguardano le categorie sportive. Mentre nei videogames l'inquadratura è pressocché identica durante tutto il video, nelle categorie sportive non lo è affatto.

Nelle categorie con una MAP più alta, Fc7 restituisce un migliore risultato rispetto al layer Prob, inoltre DeepSentiBank risulta essere più preciso rispetto alla rete Vgg.

GMM 16	Moba	Fps	Basket	Calcio
Vgg Prob	0,292	0,200	0,097	0,208
Vgg Fc7	0,349	0,221	0,043	0,158
DeepSentiBank Prob	0,353	0,214	0,077	0,183
DeepSentiBank Fc7	0,389	0,200	0,06	0,200
Vgg + DeepSentiBank Prob	0,330	0,200	0,081	0,190
Vgg + DeepSentiBank Fc7	0,388	0,204	0,052	0,177

Tabella 4.3: Esperimenti con numero centri GMM=32

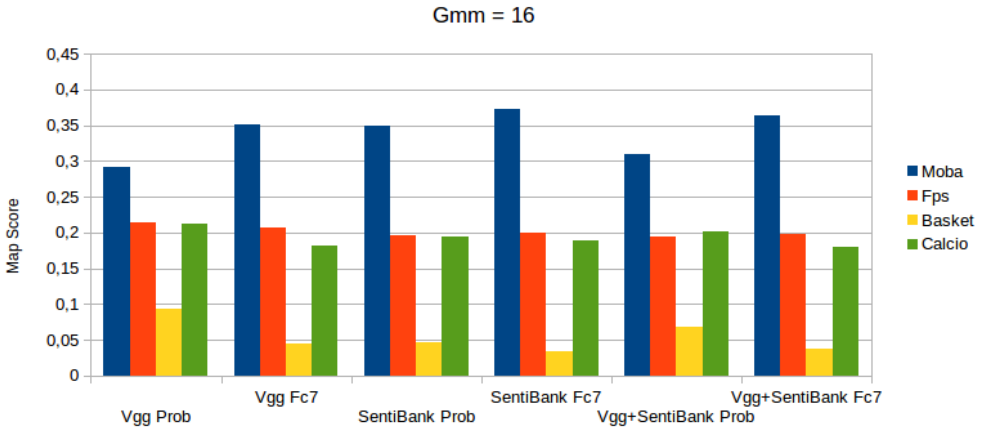

Figura 4.3: Grafico con numero centri GMM=16 con Categorie

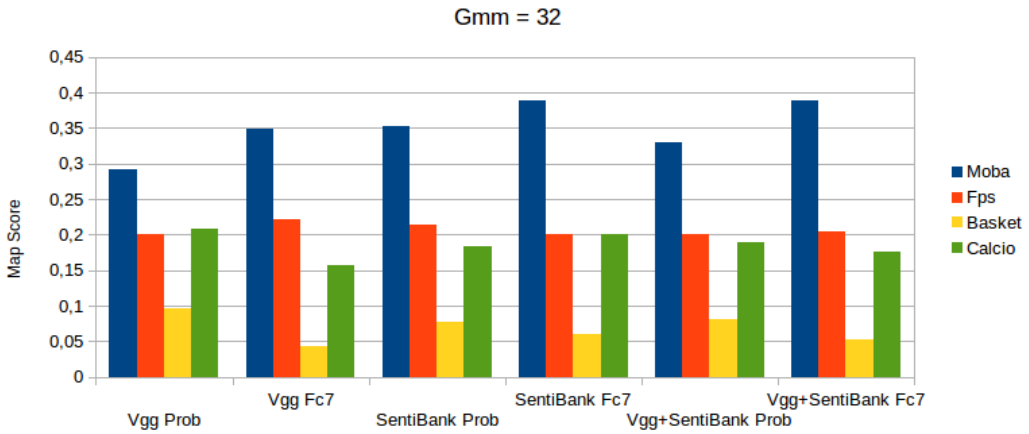

Figura 4.4: Grafico con numero centri GMM=32 con Categorie

GMM 16	Moba	Fps	Basket	Calcio
Vgg Prob	0,203	0,214	0,093	0,212
Vgg Fc7	0,291	0,207	0,045	0,181
DeepSentiBank Prob	0,352	0,197	0,047	0,195
DeepSentiBank Fc7	0,350	0,200	0,034	0,189
Vgg + DeepSentiBank Prob	0,372	0,194	0,069	0,201
Vgg + DeepSentiBank Fc7	0,309	0,198	0,037	0,180

Tabella 4.4: Esperimenti con numero centri GMM=16

4.1 Sviluppi futuri

- Allo stato attuale, il sistema di Information Retrieval fornisce solamente la similarità tra video. Si potrebbe implementare un sistema di raccomandazione che, oltre a decodificare il contenuto del video, tenga anche presente gli interessi dell'utente che utilizza il sistema IR, quindi analizzare le eventuali valutazioni che un utente potrebbe esprimere riguardo ad una serie di video precedentemente visionati.
- Sicuramente un maggior numero e una maggior cura della scelta dei dati forniti in fase di train riuscirebbe a migliorare la precision in maniera più soddisfacente, in praticolare per le categorie sportive.

Bibliografia

[1] Francesco Gelli: *Sistemi scalabili di clustering per la raccomandazione di video e l'espansione della conoscenza basati sulla profilazione semantica degli utenti di un so cial network*, Università degli studi di Firenze, Novembre, 2012.

[2] Daniele Maddaluno: *Sistemi scalabili di raccomandazione per il browsing video basati sulla profilazione semantica degli utenti di un social network*, Università degli studi di Firenze, Novembre, 2012.

[3] Saverio Meucci: *Sistemi di raccomandazione video basati su mappe di salienza e annotazioni semantiche automatiche e manuali*, Università degli studi di Firenze, Febbraio, 2015.

[4] Tao Chen, Damian Borth, Trevor Darrell, Shih-Fu Chang: *DeepSentiBank: Visual Sentiment Concept Classification with Deep Convolutional Neural Networks*, Tao Chen, Ottobre 2014.

[5] Jonathon Shlens: *A Tutorial on Principal Component Analysis*, Google Research, Aprile 2014

[6] Haralambos Marmanis, Dmitry Babenko: *Algorithms of the Intelligent Web* Haralambos Marmanis, Dmitry Babenko, Maggio 2009

Ringraziamenti

Ringrazio il Professor Marco Bertini, relatore del progetto che mi ha sostenuto per tutto la durata del lavoro e dedicato tantissimo del suo tempo prezioso. La sua competenza e le sue conoscienze sono state fondamentali per la buona riuscita del progetto.
Esprimo, inoltre, gratitudine al Dott. Andrea Ferracani per la disponibilità e l'incoraggiamento con cui mi ha supportato negli ultimi quattro mesi.

Desidero inoltre ringraziare l'intero team del Micc per la disponibilità e accoglienza che mostra agli studenti, mettendo a dispozione i propri spazi e attrezzature.

Un ringraziamente dovuto va ai miei genitori che hanno finanziato i miei studi e alleviato il mio stress durante tutta la durata del lavoro.

Non posso sicuramente dimenticare di ringraziare mio fratello Samuele, colui che ha piantato in me il seme della passione per il mondo dell'informatica quando ero più piccolo e che ancora tutt'oggi mi sostiene.

Per ultimo, ma non per questo meno importante, un ringraziamento speciale alla dolcissima Lavinia De Divitiis che è stata il mio angelo custode. Ogni volta che avevo bisogno di lei, era sempre al mio fianco.