Tehnička dokumentacija

Uvod

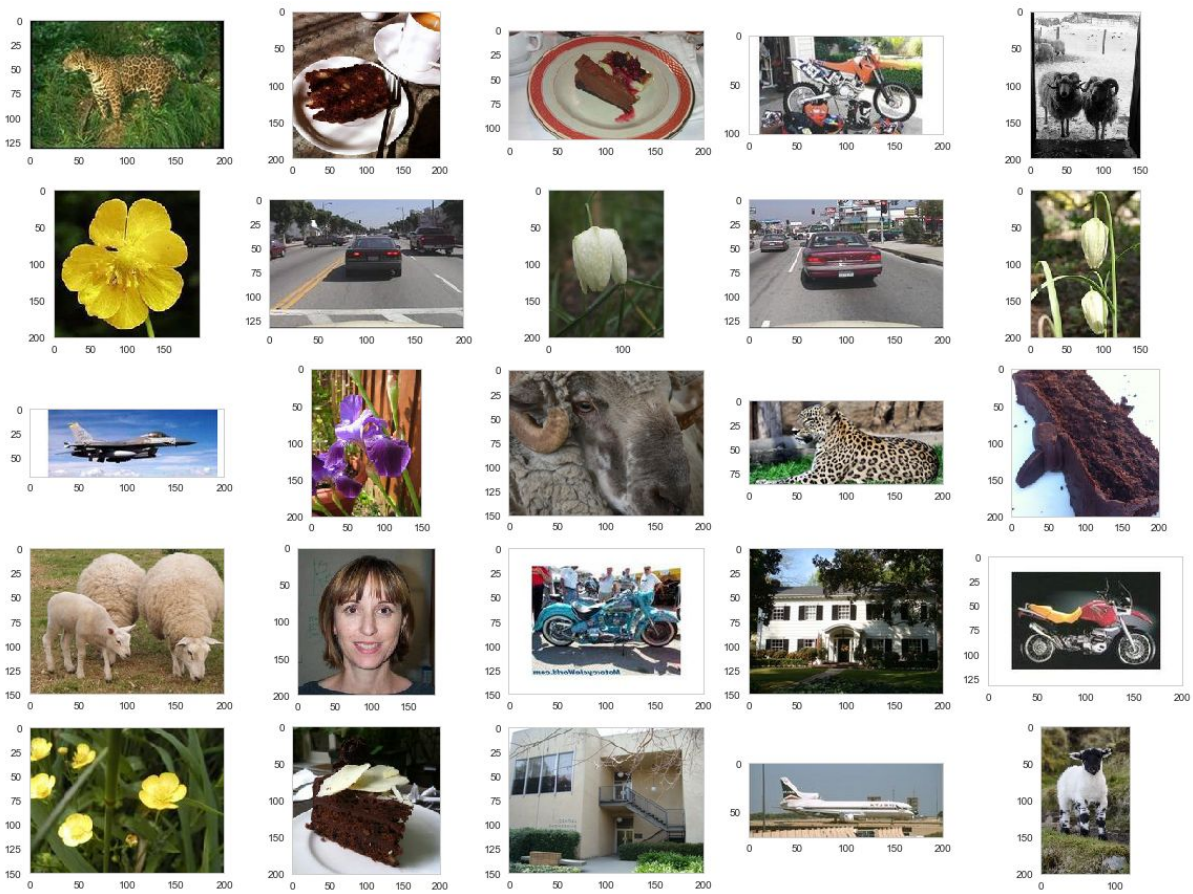
U ovom poglavlju opisan je ovogodišnji problem te analiziran sadržaj dobivenog skupa podataka.

Opis problema

Tema natjecanja je **grupiranje slika** prema njihovom sadržaju. Za zadani neoznačeni skup podataka potrebno je odrediti “optimalan” **broj grupa** i **dodijeliti svaku sliku nekoj grupi** ovisno o njenom sadržaju. Pritom algoritam ne smije imati nikakvu dodatnu informaciju o pripadnosti slika pojedinim kategorijama.

Skup podataka

Skup podataka sastoji se od 6889 slika koje su međusobno različitih dimenzija i formata. Analizom slika utvrđeno je da postoji **desetak** vidljivo različitih semantičkih sadržaja, npr. avioni, motori, građevine, lica, kolači, cvijeće, …
Skup podataka sadrži i važno svojstvo da je u svakoj slici jedan objekt dominantan i vidljivo istaknut. Uzorak iz skupa podataka prikazan je na slici 1.



Slika 1: Uzorak iz skupa podataka

Tehnologija

U ovom poglavlju detaljnije su opisane i obrazložene korištene tehnologije.

Programski jezik

Programski jezici koji se najčešće koriste pri radu s neuronskim mrežama su Python, R, Matlab i C++. U ovom projektu odabran je jezik **Python** jer je vrlo jednostavan za korištenje i omogućuje brzu izradu prototipova. Također, velik skup dodatnih biblioteka čini ga pogodnim za rad s neuronskim mrežama i općenito strojnim učenjem.

Dodatne biblioteke

U nastavku su opisane biblioteke otvorenog koda (engl. *open-source*) korištene u projektu.

Scikit-learn
Biblioteka za strojno učenje koja pruža implementacije brojnih modela za klasifikaciju, regresiju i grupiranje. Dodatno, omogućuje predobradu, odabir modela, redukciju dimenzionalnosti i drugo.
Biblioteka je u projektu korištena za **analizu glavnih komponenti** (engl. *Principal Component Analysis - PCA*), **grupiranje** i **odabir optimalnog broja grupa**.

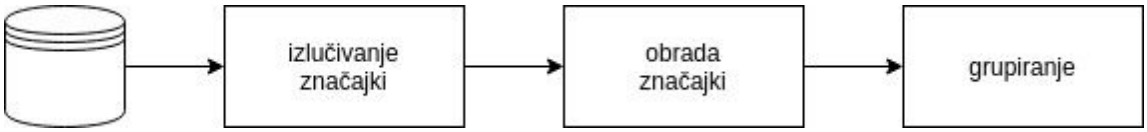
Scikit-image, OpenCV
Biblioteke koje pružaju algoritame za obradu slika.
U ovom projektu korištene su za učitavanje i manipulaciju različitih formata slikovnih datoteka koje se javljaju u ulaznom skupu podataka (*jpg*, *png*, *tif*, *gif*).

Tensorflow
Biblioteka razvijena od strane *Google Brain Teama* za rad s neuronskim mrežama. Pruža mogućnost učenja i izvođenja modela na grafičkim karticama.
Biblioteka je u projektu korištena za izlučivanje značajki iz ulaznih podataka. Osim *čistog* Tensorflowa, korišten je i **Keras**. Radi se o aplikacijskom sučelju visoke razine apstrakcije nad Tensorflowom (i nad Theanom) koje olakšava rad s neuronskim mrežama.

Koraci rješenja

Izrada rješenja može se podijeliti na tri bitna koraka (slika 2):

1. Izlučivanje značajki
2. Obrada značajki
3. Grupiranje vektora značajki



Slika 2: Dijagram koraka rješenja

U nastavku slijedi detaljan opis svakog koraka.

Izlučivanje značajki

Glavna ideja ovog koraka je koristiti **Transfer learning** metodu tako da prethodno naučenu neuronsku mrežu iskoristimo u svrhu izlučivanja značajki.

Izlazi odabranog sloja mreže predstavljaju značajke slike. Razlog za to slijedi iz načina na koji duboke konvolucijske neuronske mreže rade. Niži slojevi uče jednostavne filtre (primjerice detekcija bridova i sl.), dok viši slojevi prepoznaju složenije strukture. To je posljedica pretpostavke modela da su značajke slika hijerarhijski građene, odnosno da se složenije značajke mogu rastaviti na jednostavnije. Konačno, potpuno povezani slojevi mreže iz takvih složenijih značajki određuju kojoj grupi pojedini primjer pripada.

U ovom projektu korištene su mreže naučene na **ImageNet** skupu podataka koji se sastoji od otprilike milijun slika podijeljenih u 1000 grupa. Budući da su mreže učene na velikim skupovima podataka s različitim grupama, takve bi mreže u svojim posljednjim slojevima (bez završnih potpuno povezanih) trebale sadržavati kvalitetne složene značajke.

Predobrada slika

Budući da koristimo mrežu prethodno naučenu na ImageNetu, potrebno je napraviti i predobradu (engl. *preprocessing*) slika kako je to učinjeno i pri njenom učenju. Predobradom se postiže:

- skaliranje slika na traženu rezoluciju, i
- ugađanje raspona vrijednosti piksela.

Modeli

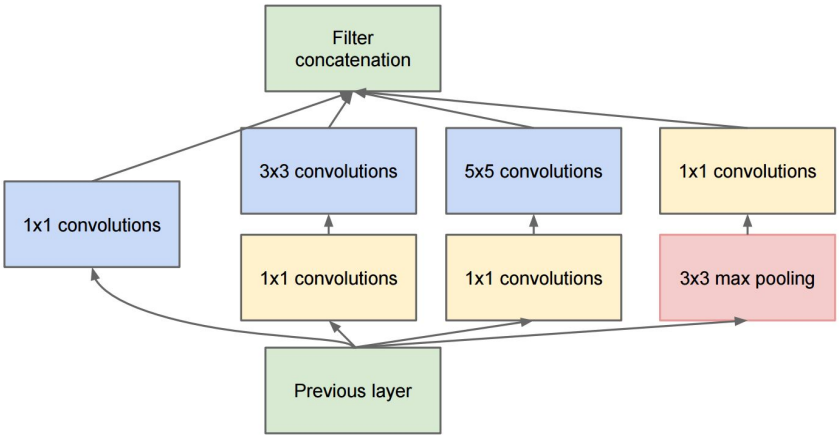
U ovom koraku isprobano je nekoliko modela, a značajke su izlučivane na različitim dubinama mreže. Cilj je bio usporediti kvalitetu rješenja i odabrati najbolje. Isprobani modeli prikazani su u tablici 1. U tablici je označena i dimenzija slike koju model prima na ulazu te broj značajki koji se korištenjem tog modela dobiva.

Inception i **ResNet** su jedne od najsuvremenijih i najmoćnijih arhitektura. Svaka od njih ima jedinstvenu ideju koja doprinosi klasičnoj arhitekturi konvolucijskih mreža koje se sastoje od jednostavnog uzastopnog povezivanja konolucijskih slojeva i slojeva sažimanja.

Tablica 1: Modeli korišteni pri izlučivanju značajki

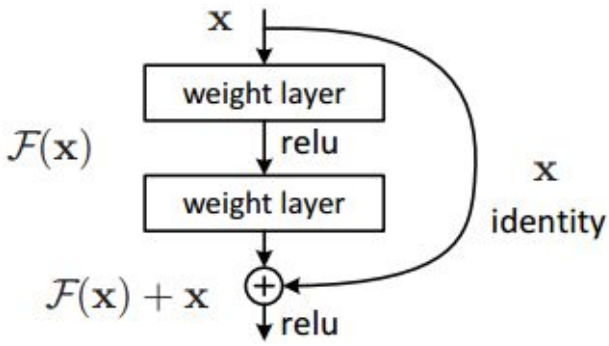
model	*ulazna dimenzija*	*broj značajki*
Inception v3	229x229x3	2048
Inception v4	229x229x3	16136
ResNet50	224x224x3	2048

Inception blok ima više konvolucijskih jezgri raznih veličina što doprinosi nelinearnosti i izlučivanju značajki različitih razina. Konvolucije s malim jezgrama izvlače finije detalje iz slike, dok one s većim jezgrama i većim receptivnim poljem izvlače informacije ovisne o većem kontekstu slike. Primjer *inception bloka* prikazan je na slici 3.



Slika 3: Primjer *inception bloka*

S druge strane, rezidualne mreže u svojim slojevima koriste jednostavne konvolucije i ne idu u širinu kao što je slučaj kod *inceptiona*. Ideja je da se određeni ulaz propagira ne samo u sljedeći sloj, nego i u neki nakon njega. To znači da određeni sloj na svom ulazu ne dobiva samo izlaz prethodnog sloja, već i nekog od nižih te ih kombinira operacijom zbrajanja. Ovaj pristup omogućuje treniranje dubljih mreža jer spomenute rezidualne veze omogućavaju propagaciju gradijenta unatrag kroz mrežu bez iščezavanja gradijenata. Primjer *resnet bloka* prikazan je na slici 4.



Slika 4: Primjer *resnet bloka*

Obrada značajki

Normalizacija

Normalizaciju koristimo kako bi eliminirali različite skale po komponentama koje bi mogle dovesti do pogrešaka pri izračunu sličnosti između dva vektora značajki.
U projektu je korištena **StandardScaler** metoda iz *Sklearn* paketa koja komponente svodi na normalnu jediničnu razdiobu.

Redukcija dimenzionalnosti

Kao što je već ranije pokazano, prilikom izlučivanja značajki na različitim dubinama mreže dobiveni su vektori značajki različitih veličina. To su uglavnom bili visoko dimenzionalni vektori, stoga su oni preslikani u prostor niže dimenzionalnosti. Cilj je bio dobiti sažetu reprezentaciju vektora koja u sebi sadržava bitne informacije o sadržaju slike.

Analiza glavnih komponenti

Analiza glavnih komponenti (engl. *PCA*) je metoda za redukciju podataka koja reducira velik broj varijabli na mali broj kompozitnih varijabli.
PCA je u projektu korištena za preslikavanje vektora značajki u prostor niže dimenzionalnosti, a kao parametar prima željenu konačnu dimenziju vektora.

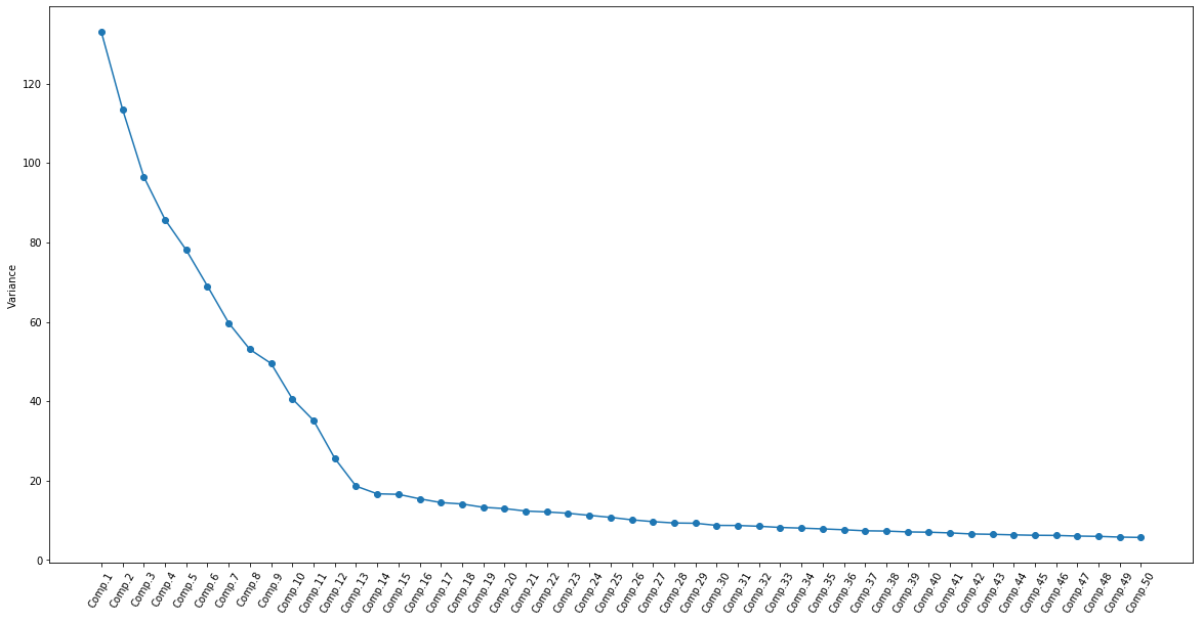
Odabir dimenzije

Odabir konačne dimenzije vektora značajki ne smije biti slučajan, nego teorijski potkrepljen. U svrhu izračuna dimenzije korišteno je nekoliko metoda, a rezultati koji su njima dobiveni detaljnije su opisani u nastavku.

Kaiser metoda
Metoda pronalazi sve svojstvene vrijednosti korelacijske matrice. Konačna dimenzija tada je jednaka broju svojstvenih komponenti s vrijednostima većim od 1.

Postatak objašnjenje varijance
Ova metoda (engl. *Percentage of variation explained)* ostavlja 80% varijance u podacima jer u tom postotku postoji dovoljno deskriptivnih svojstava podataka bez velikog utjecaja šuma.

Scree test
Metoda iscrtava svojstvene vrijednosti opadajuće prema veličini u odnosu na broj komponenti. Broj značajnih komponenti nalazi se na koljenu tako iscrtanog grafa.



Slika 5: Graf Scree testa za vektore značajki dobivene u projektu

U projektu su isprobane sve tri navedene metode. Graf dobiven Scree testom prikazan je na slici 5. Metodom koljena određeno je da je broj značajnih komponenti za ovaj izračun jednak 13. Ostale dva izračuna su automatskog tipa. Svi rezultati prikazani su u tablici 2.

Tablica 2: Izračun konačne dimenzije vektora značajki

metoda	*tip metode*	*dimenzija*
Kaiser metoda	automatska	22
Scree test	ručna	13
Postotak objašnjene varijance	automatska	156

Ručnim pregledom grupa ustanovljeno je da su sva tri odabira konačne dimenzije vektora značajki dala slične rezultate. Dodatno, isprobane su i vrijednosti manje od 13 koje su dale uočljivo slabije rezultate.

U projektu je stoga odabrana vrijednost 13 koja se pokazala dovoljno ekspresivnom za dani skup podataka. Za buduće primjene ostavljena je mogućnost izbora između automatske metode (*Kaiser metoda*) i ručne metode (*Scree test*).

Grupiranje vektora značajki

Nakon izlučivanja značajki, završni korak je **grupiranje vektora značajki** u grupe prema sličnosti, na način da vektori unutar iste grupe budu što bliži, a vektori različitih grupa što udaljeniji. Pritom je potrebno odrediti i **optimalan broj grupa**.

Algoritmi grupiranja

U svrhu grupiranja sličnih vektora značajki korištene su dvije metode: algoritam k-srednjih vrijednosti i hijerarhijsko grupiranje. Obje su metode dostupne u programskom paketu *Sklearn*, a detaljnije su opisane u nastavku.

Algoritam k-srednjih vrijednosti (engl. *K-means*)
Model grupira podatke u *k* različitih grupa podjednakih varijanci, minimizirajući uvjet poznat kao inercija ili srednje kvadratno odstupanje unutar grupe. Model zahtijeva unaprijed zadani broj grupa.
Ovaj algoritam vrlo se često koristi u problemima grupiranja. U ovom projektu odabran je zbog pogodne vremenske i memorijske složenosti. To osobito dolazi do izražaja u slučaju velikog broja primjera.

Hijerarhijsko grupiranje (engl. *Hierarchical clustering*)
Hierarhijsko grupiranje grupira podatke spajajući manje grupe u sve veće. Hijerarhija je često predstavljena stablastom strukturom zvanom **dendrogram**. Korijen stabla je jedinstvena grupa sa svim podacima, dok listovi predstavljaju grupe sa samo jednim primjerom.
Ovo grupiranje odabrano je kako bi dobili hijerarhijsku strukturu koja otkriva relacije između grupa. Ta struktura nam pomaže odrediti koje su grupe međusobno slične te na koje se podgrupe neka grupa može podijeliti.

Optimalan broj grupa

Budući da broj grupa nije unaprijed poznat, potrebno ga je na smisleni način dobiti iz skupa podataka. Analizom zadanih slika utvrđeno je da se prema sadržaju razlikuje **desetak** grupa, ali rješenje mora raditi i na proizvoljnom skupu podataka. Stoga su za izračun optimalnog broja grupa korištene pomoćne statističke metode opisane u nastavku.

Silhouette metoda
Kvalitetu broja grupa određuje pomoću dviju mjera: srednja udaljenost primjera od ostalih primjera iste grupe (*a*) i od primjera najbliže grupe (*b*). Kvaliteta se zatim određuje formulom:

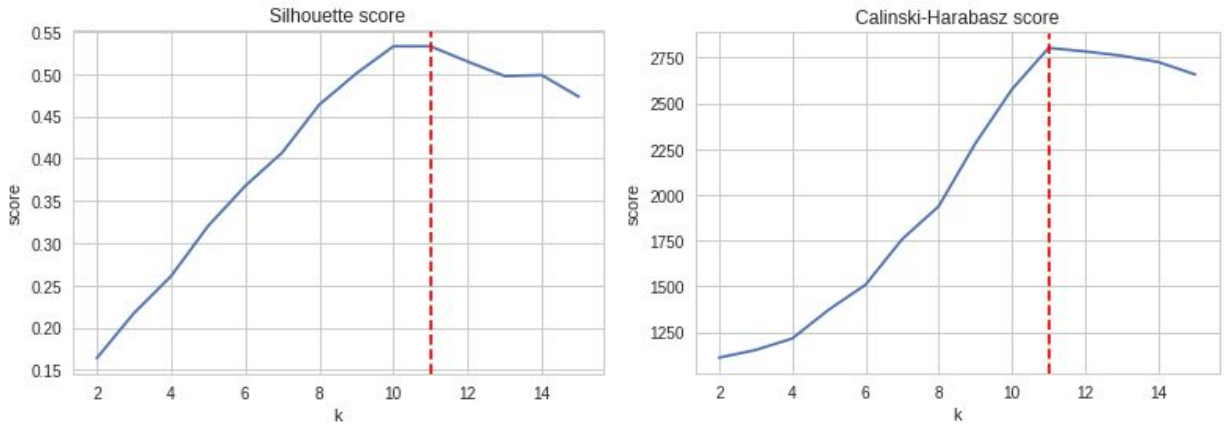
$$\frac{b-a}{max(a,b)}$$

Izraz poprima vrijednost u intervalu [**-1**, **1**], gdje **1** označava da su grupe kompaktne i međusobno jako udaljene, **0** da se primjeri pojedinih grupa nalaze na granici, a **-1** da se grupe preklapaju. Ova metoda prilagođena je algoritmu *k*-srednjih vrijednosti.

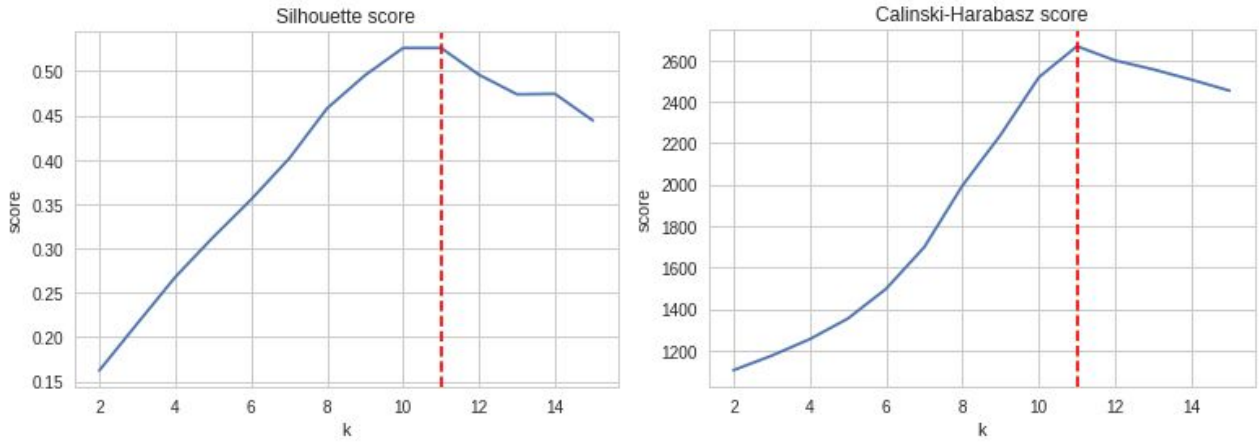
Calinski-Harabaz metoda
Vrijednost predstavlja omjer raspršenosti unutar grupe i raspršenosti između grupa. Najbolji broj grupa je onaj za kojeg metoda poprima najveću vrijednost.

Stability metoda
Računa stabilnost modela grupiranja za određeni broj grupa [1]. Ulazni podaci se dijele na dva podskupa, nakon čega se nad svakim podskupom napravi grupiranje i računa sličnost između dobivenih grupa. Postupak se ponavlja više puta.

Gap statistika
Metoda određuje broj grupa uspoređujući inerciju unutar grupa nastalih grupiranjem stvarnih podataka s inercijom grupiranja slučajnih podataka [2].



Slika 6: Određivanje optimalnog broja grupa za algoritam *k*-srednjih vrijednosti



Slika 7: Određivanje optimalnog broja grupa za hijerarhijsko grupiranje

Zbog velike računske složenosti ostalih metrika, korištene su prve dvije metode (*Silhoutte* i *Calinski-Harabaz*). Obje metode pokrenute su za algoritam *k*-srednjih vrijednosti i hijerarhijsko grupiranje. Dobiveni grafovi prikazani su na slikama 6 i 7.

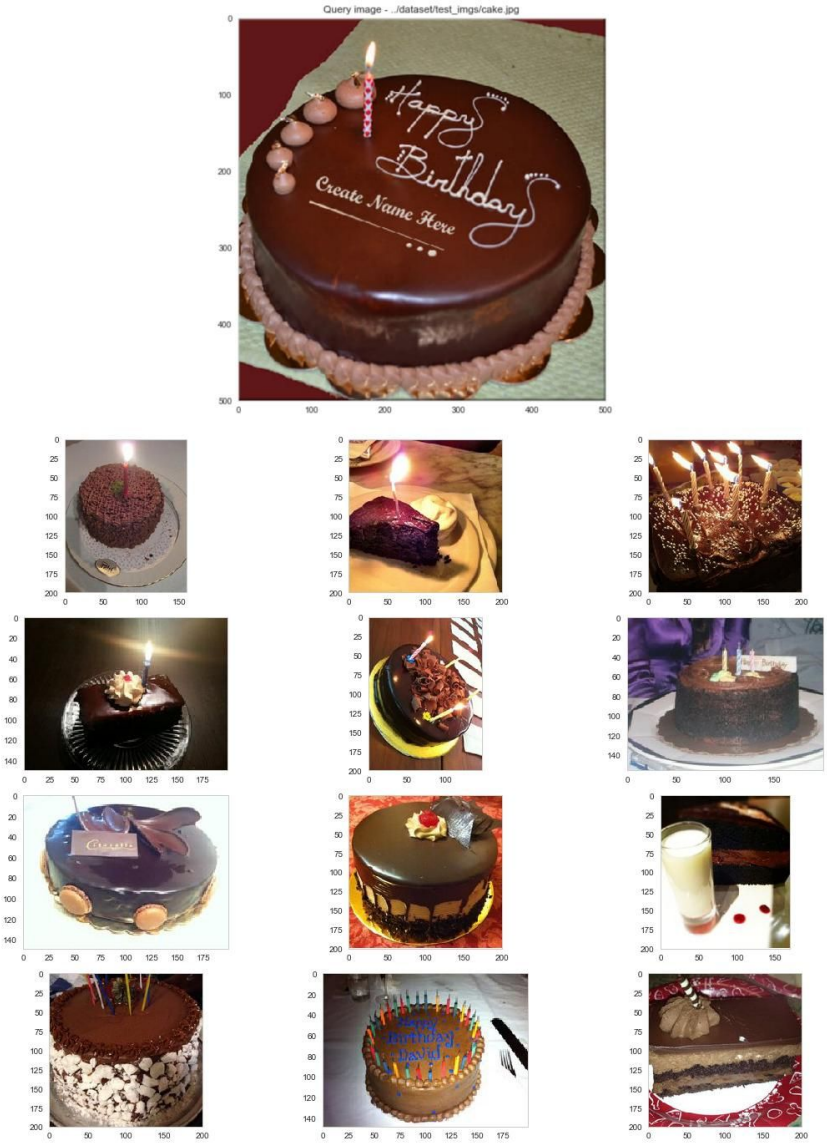
Sve metode za oba korištena algoritma daju jednak optimalan broj grupa - 11.

Evaluacija

Analiza kvalitete izlučivanja značajki

Budući da su slike nakon izlučivanja značajki predstavljene vektorima značajki, prije daljnjeg rada na odabiru broja grupa i samog grupiranja, analizirana je kvaliteta dobivene vektorske reprezentacije. Za izlučene značajke možemo reći da su dobre ako slične slike iz ulaznih podataka imaju i slične vektorske reprezentacije.

U tu svrhu implementirana je jednostavna metoda koja za zadanu sliku vraća *K* slika iz ulaznog skupa podataka s najsličnijim vektorima značajki. Pritom su korištene definirane funkcije sličnosti: kosinusna sličnost i recipročna vrijednost euklidske udaljenosti vektora značajki.



Slika 8: Torta i 16 njoj u vektorskom prostoru najsličnijih slika

Slika 9: Zrcaljeni avion i 16 njemu u vektorskom prostoru najsličnijih slika

Prikaz dobivenih rezultata za dva primjera nalazi se na slikama 8 i 9. U prvom redu nalazi se slika odabrana iz ulaznog skupa podataka, a ispod nje nalazi se 16 njoj u vektorskom prostoru najsličnijih slika. Iz danih je prikaza vidljivo da su značajke kvalitetno izlučene, jer slične vektorske reprezentacije predstavljaju i slične slike.

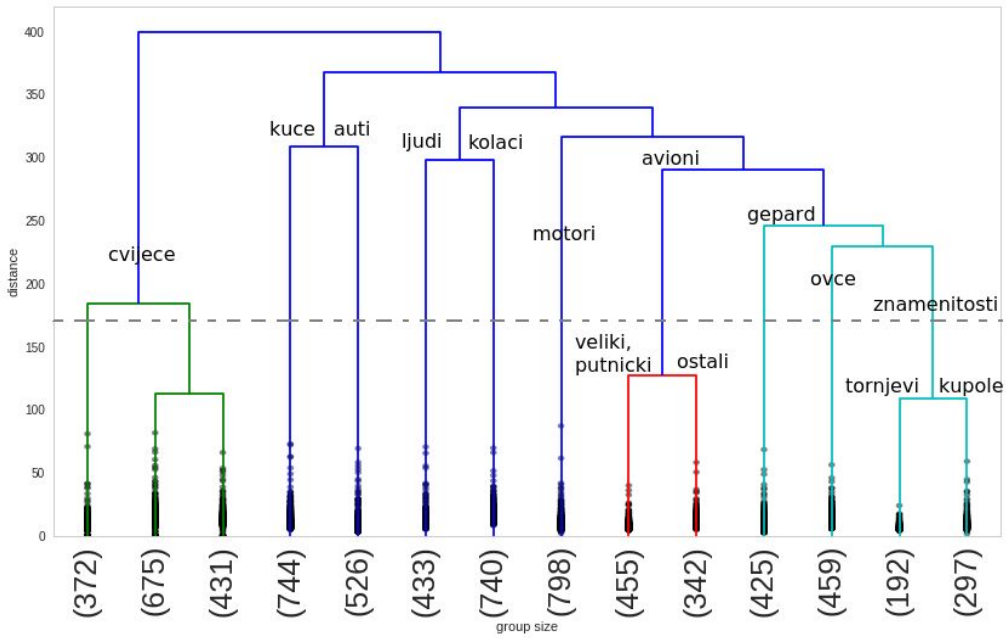
Rezultati na zadanom skupu podataka

Kao što je ranije opisano, prilikom izrade rješenja isprobani su različiti modeli i metode. U tablici 3 prikazani su rezultati evaluacija svakog rješenja. Za mjeru sličnosti korištena je **kosinusna** sličnost. Tablica prikazuje o kojem se modelu radi, koliko iznosi prosječna sličnost unutar dobivenih grupa, prosječna sličnost između različitih grupa i apsolutni omjer ta dva broja.

Tablica 3: Evaluacija različitih rješenja na zadanom skupu podataka

model	*unutar grupa*	*između grupa*	*omjer*
ResNet50 + PCA(22) + KMeans	0.394507	-0.036482	10.81
ResNet50 + PCA(22) + HC	0.390849	-0.036097	10.83
ResNet50 + PCA(13) + KMeans	0.788341	-0.076315	10.33
ResNet50 + PCA(13) + HC	0.786594	-0.076002	10.35
Inception v3 + PCA(22) + KMeans	0.421672	-0.039564	10.66
Inception v3 + PCA(22) + HC	0.422741	-0.039374	10.74
Inception v3 + PCA(13) + KMeans	0.800979	-0.077094	10.39
Inception v3 + PCA(13) + HC	0.797859	-0.076517	10.43
Inception v4 + PCA(281) + KMeans	0.304782	-0.026247	11.61
Inception v4 + PCA(281) + HC	0.708633	-0.067622	10.48
Inception v4 + PCA(18) + KMeans	0.708633	-0.067622	10.48
Inception v4 + PCA(18) + HC	0.708141	-0.067339	10.52

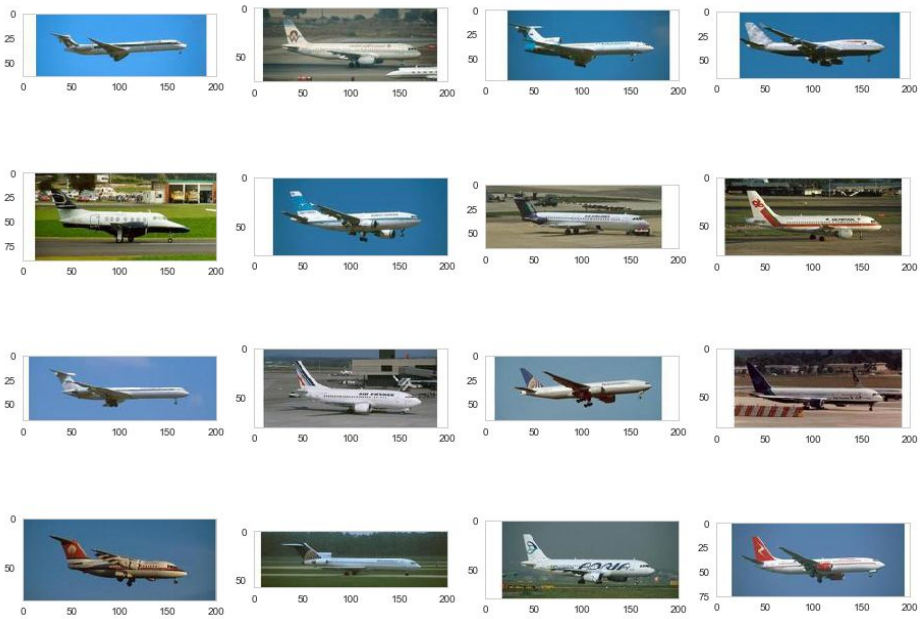
Rezultati iz tablice pokazuju da sva isprobana rješenja daju kompaktne grupe s međusobno sličnim primjerima te da su grupe međusobno jasno odvojene. Taj zaključak je dodatno potvrđen vizualnim prolaskom kroz stvorene grupe. Budući da su svi modeli dali približno jednake rezultate, u daljnjem radu korišten je *Inception v3* s PCA preslikavanjem u dimenziju 22 koja je automatski odabrana *Kaiser* metodom.



Slika 10: Dendrogram dobivenih grupa

Na slici 10 dendrogramom su prikazani detaljniji rezultati dobivenim hijerarhijskim grupiranjem. Iz slike su vidljivi hijerarhijski odnosi među grupama. Isprekidanom sivom crtom označena je dubina podjele na 11 grupa. Broj primjera pojedinih grupa označen je brojevima u zagradi.

Zanimljivo je uočiti da se odsijecanjem na većoj dubini dobiva finija podjela grupa. Tako se avioni dijele na putničke i ostale vrste, znamenitosti na kupole i tornjeve, a cvijeće na podvrste. Detaljnija podjela aviona prikazana je na slikama 11 i 12.



Slika 11: Putnički avioni

Slika 12: Ostale vrste aviona

Korištenjem metode *t-SNE*, značajke i labele dobivene grupiranjem prikazane su u 2D prostoru. Na slici 13 vide se jasno odvojene grupe kao i naznake njihovih podgrupa.

Slika 13: Prikaz dobivenih grupa u 2D prostoru

Rezultati na označenim skupovima podataka

Budući da je izrazito teško evaluirati točnost nenadziranog grupiranja, odabrano rješenje testirano je i na javno dostupnim označenim skupovima podataka. Kako bi se omogućio smisleni izračun kvalitete, bilo je potrebno nakon grupiranja, a prije samog izračuna, napraviti labeliranje grupa.

Svim primjerima unutar grupe dodijeljena je labela koja odgovara većinskoj originalnoj labeli primjera u toj grupi. S tako dobivenim labelama izračunate su performanse modela: točnost, preciznost, odziv i F1 mjera. U slučaju višerazrednog grupiranja, korištene su makro mjere koje računaju prosjek mjera po svim grupama. Osim spomenutih mjera, korišten je i

prilagođeni rand indeks (engl. *Adjusted Rand Index - ARI*) između dobivenih i originalno označenih labela.

Rand indeks opisuje mjeru sličnosti između dva grupiranja uzimajući u obzir broj parova primjera u istim grupama i broj parova primjera u različitim grupama. *Prilagođeni rand index* određuje se iz *Rand indeksa* pomoću sljedeće formule:

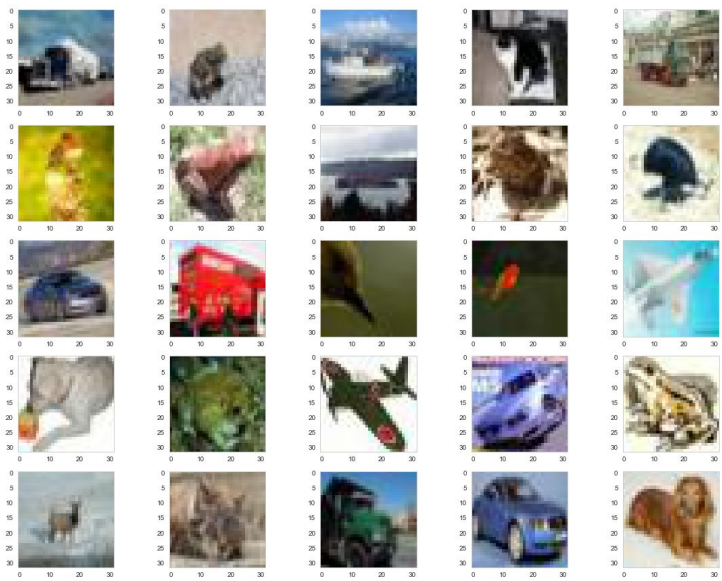
$$ARI = \frac{RI - RI_{očekivani}}{RI_{max} - RI_{očekivani}}$$

Nezavisna slučajna grupiranja imat će vrijednost približno jednaku **0**, dok će grupiranja ista do na redoslijed labela imati vrijednost jednaku **1**.

ARI nažalost nije idealna mjera jer kažnjava rezultate u kojima je neka grupa razbijena na više podgrupa, u kojima jedno od grupiranja ima finiju granulaciju ili ako su grupiranja napravljena po potpuno različitim kriterijima. Zbog toga je, kako bi rezultat grupiranja bio usporediv sa stvarnim grupama, broj grupa u skupu podataka unaprijed zadan prema stvarnom broju grupa.

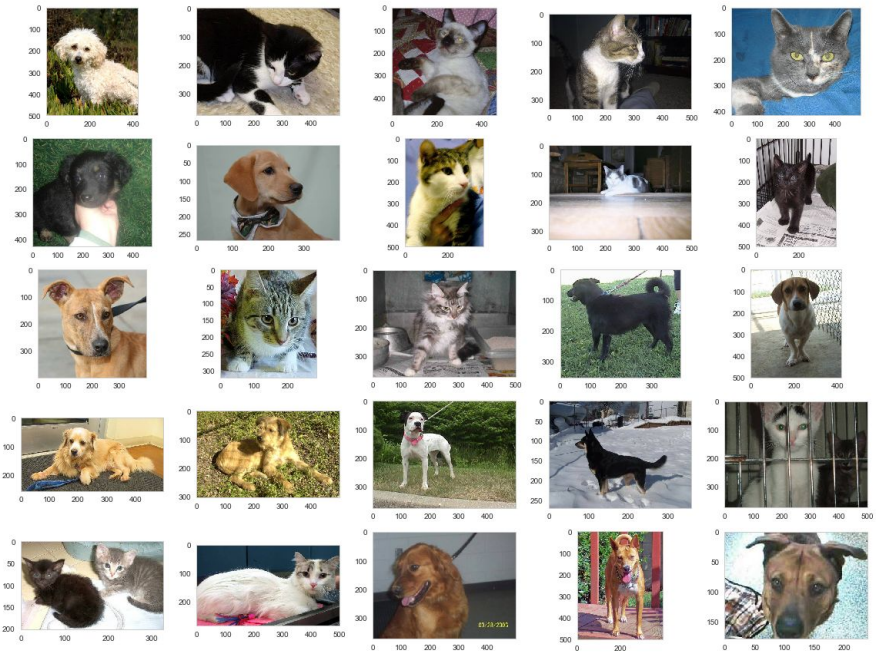
U nastavku su opisani korišteni označeni skupovi podataka.

CIFAR-10
Skup podataka s 50000 slika dimenzija 32x32 podijeljenih u 10 grupa s otprilike 5000 primjera po grupi. Uzorak slika iz skupa podataka prikazan je na slici 14.



Slika 14: Uzorak iz skupa *CIFAR-10*

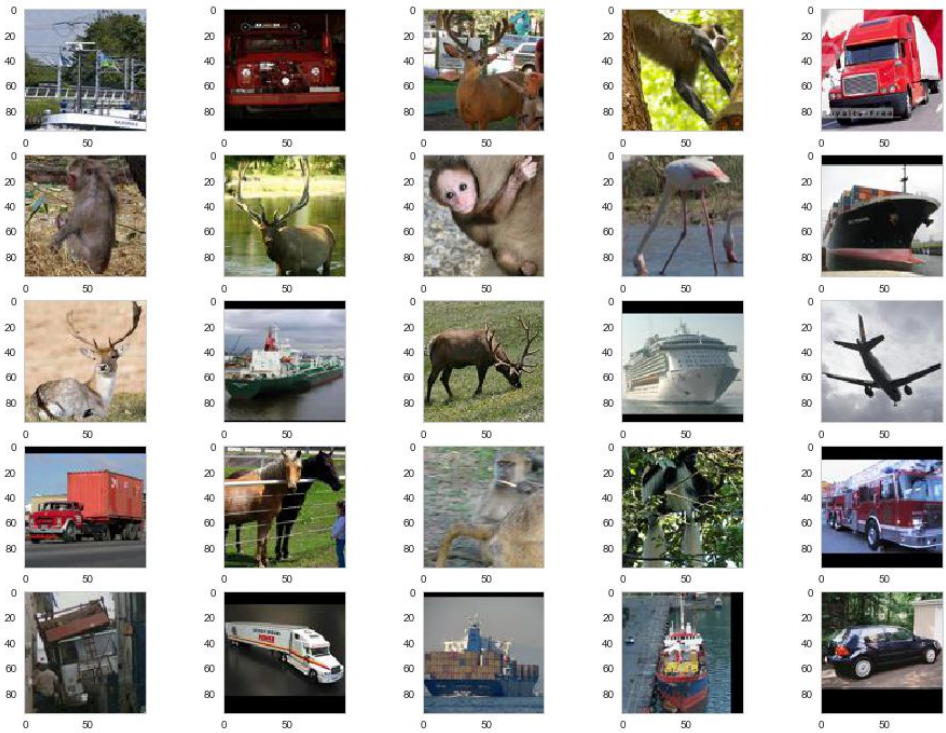
Dogs vs. cats
Skup podataka s *Kaggle* natjecanja koji sadrži 25000 slika različitih rezolucija i dimenzija podijeljenih u 2 grupe (*mačke* i *psi*). Korišten je kao primjer jednostavnijeg skupa podataka. Uzorak slika prikazan je na slici 15. Dostupan je na https://www.kaggle.com/c/dogs-vs-cats.



Slika 15: Uzorak iz skupa *Dogs vs. cats*

STL-10

Skup podataka s 5000 slika dimenzija 96x96 podijeljenih u 10 grupa s 500 primjera po grupi. Inspiriran je *CIFAR-10* skupom podataka, ali su slike veće rezolucije. Uzorak slika prikazan je na slici 16. Skup je dostupan na https://cs.stanford.edu/~acoates/stl10/.



Slika 16: Uzorak iz skupa *STL-10*

Rezultati izvođenja odabranog rješenja na gore spomenutim označenim skupovima podataka prikazani su u tablici 4.

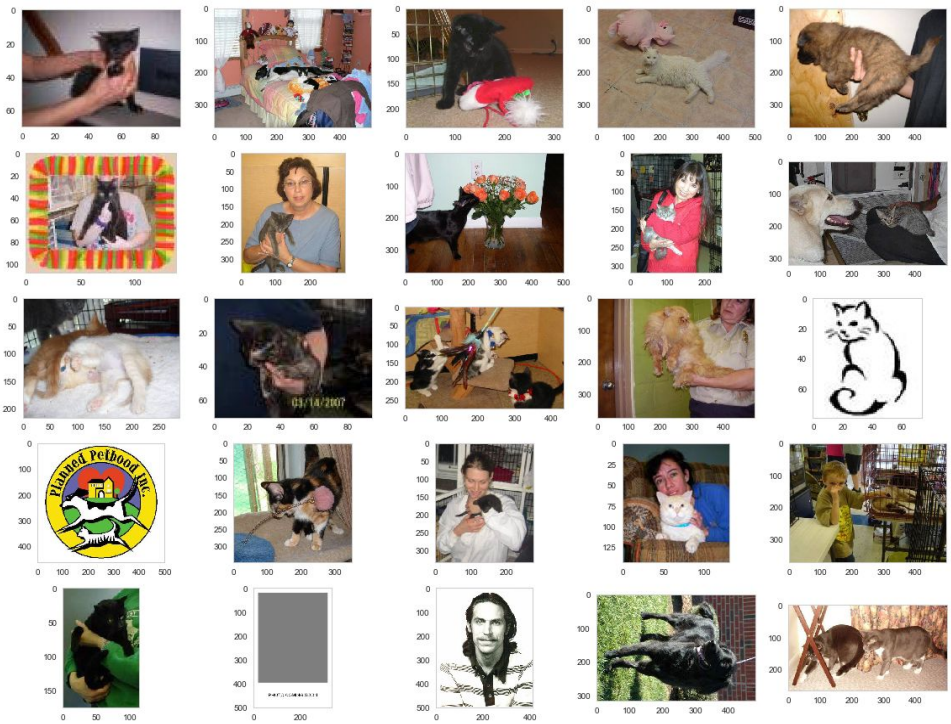
Tablica 4: Evaluacija odabranog rješenja na označenim skupovima podataka

skup podataka	*točnost*	*preciznost*	*odziv*	*F1*	*ARI*
CIFAR-10	0.5518	0.5218	0.5518	0.5310	0.3116
STL-10	0.9074	0.9074	0.9074	0.9104	0.7993
Dogs vs. cats	0.99112	0.9912	0.9911	0.9911	0.9648

Očekivano, *CIFAR-10* skup podataka pokazao se teškim problemom za nenadzirano učenje jer su slike malih dimenzija, a grupe međusobno slične (konji-jeleni, automobili-kamioni). Zbog toga su dobiveni rezultati koji su malo iznad 50% zadovoljavajući.

Kao glavni razlog lošijih rezultata na skupu *CIFAR-10* pokazala se upravo mala dimenzija (32x32) jer su na sličnom skupu podataka *STL-10*, koji također ima nekoliko međusobno sličnih grupa, postignuti značajno bolji rezultati.

Dogs vs. cats skup podataka pokazao se jednostavnijim problemom. Rezultati grupiranja su vrlo dobri, a do pogreške je došlo na svega 222 od ukupno 25000 slika. Analizom pogrešaka ustanovljeno je da su to većinom slike na kojima se istovremeno nalaze i psi i mačke, ljudi ili je naprosto originalni skup pogrešno označen. Uzorak pogrešno grupiranih slika prikazan je na slici 17.



Slika 17: Uzorak *Dog vs. cats* pogrešno grupiranih slika

Zaključak

Pri rješavanju ovog problema razmatrani su različiti pristupi, a odabran je onaj koji se već više puta pokazao uspješnim. Namjera je bila detaljno analizirati odabrani pristup, dublje ga upoznati i potvrditi njegovu kvalitetu na zadanim podacima.

Analizom izlučenih značajki potvrđena je opravdanost korištenja naučene neuronske mreže u svrhu izlučivanja značajki. Osim toga, u projektu je prikazan rezultat više različitih metoda za određivanje broja bitnih dimenzija vektora značajki i određivanje optimalnog broja grupa. Provedena su i dva različita algoritma grupiranja.

Vizualnim pregledom dobivenih grupa ustanovljeno je da svaka grupa predstavlja različiti objekt, odnosno da se grupe semantički razlikuju. Kvaliteta rješenja dodatno je potvrđena evaluacijom na drugim, javno dostupnim označenim skupovima podataka.

U daljnjem radu bilo bi zanimljivo umjesto konvolucijskih mreža u svrhu izlučivanja značajki koristiti i druge tipove mreža, primjerice autoenkodere ili neke generativne modele kao što su *generative adversarial networks*.

Literatura

[1] Ben-Hur, Elisseeff, Guyon: A stability based method for discovering structure in clustered data, 2002
[2] R.Tibshirani, G. Walther and T.Hastie, Estimating the number of clusters in a dataset via the Gap statistic, Journal of the Royal Statistical Society: Series (B) (Statistical Methodology), 63(2), 411-423