

PhotoBrowser: navegador/cercador d'imatges basat en algorismes de visualització “big data”

Marc López Serrano

Resum—El món actual disposa d'una immensa quantitat de dades que cada vegada va creixent. Un d'aquest tipus de dades són les fotografies que realitza cada persona, ja sigui per qüestions laborals o personals. Tant les xarxes socials com altres plataformes, disposen cada vegada més fotografies a l'any. Per poder-se adaptar als temps que arriben, és imprescindible gestionar aquesta quantitat d'informació de forma eficient i amb la capacitat de visualitzar de manera senzilla una imatge d'un àmbit específic. Per això, aquest projecte es proposa com una solució a la cerca d'imatges en una immensa quantitat de dades. Amb l'utilització d'algorismes de BigData, podem buscar una imatge concreta molt més fàcilment de cara a buscar en una base de dades molt gran. En particular, el projecte ha consistit en el desenvolupament d'una plataforma web de visualització i cerca sobre grans volums d'imatges de manera user friendly. Integra l'algorisme t-SNE (t-distributed stochastic neighbor embedding), utilitzant un gran volum d'imatges com a dades d'entrada.

Paraules clau— Arquitectura MVC, BigData, Cercador d'imatges, T-SNE, Aplicació web, HTML, PHP, JSON, Python, D3js

Abstract—The modern world has a vast amount of data that is growing more and more. One of these types of data are the pictures that makes every person, whether for work or personal issues. Both social networks and other platforms, have more and more photos per year. To be able to adapt to the time they arrive, it is essential to manage this amount of information efficiently and with the ability to visualize in a simple way a picture of a specific area. Therefore, this project is proposed as a solution to the search for images in an immense amount of data. With the use of BigData algorithms, we can look for a specific image much more easily in order to search in a very large database. In particular, the project consisted of the development of a web platform or visualizing and browsing thorough big collections of pictures in a user friendly way. It integrates the t-SNE algorithm (t-distributed stochastic neighbor embedding), using a large volum of pictures as input data.

Index Terms— MVC architecture, BigData, Photo Browser, T-SNE, Web application, HTML, PHP, JSON, Python, D3js

1 INTRODUCCIÓ

A vui dia disposem de moltes fotografies en els nostres ordinadors i dispositius mòbils, ja siguin descarregades o fetes amb la càmera. A conseqüència d'això, la societat és impulsada a la utilització de bases de dades d'imatges, però aquesta situació pot presentar una sèrie de dificultats a l'intentar buscar una imatge concreta en el gran volum d'imatges que es té. Es disposa de múltiples eines per la gestió d'aquest tipus de bases de dades, però no són molt precises en termes de cerca i visualització. Tenint en compte que també predomina el terme de BigData, cada vegada, les dades són més complexes i les imatges s'han d'adaptar a aquesta realitat vigent [1].

En aquest projecte s'ha dissenyat i desenvolupat una eina de visualització d'imatges capaç de trobar les similituds que té una imatge de mostra amb altres, extretes d'una base de dades basada en imatges. Es podrà classifi-

car les similituds amb tota les que hi ha en la base de dades o algunes especificades en un àmbit concret. Un exemple de la funcionalitat a aplicar seria similar al que utilitza Google Images, el qual mitjançant una imatge, mostra en forma de mosaic les coincidències que s'han trobat. En comparació amb l'exemple anterior, l'eina que s'ha desenvolupat visualitza els resultats extrets com núvols de punts el qual determina els punts més propers (més similituds amb la mostra) i els més distants (poca o cap similitud).

El treball ha consistit a analitzar diferents components de software per a la visualització de grans volums de dades (big data) i en particular l'algorisme t-SNE. S'ha dissenyat i desenvolupat una aplicació basada en una plataforma web que permeti importar diferents conjunts d'imatges, i definir diversos criteris de visualització.

La resta d'aquest document, s'estructura de la manera següent:

- *E-mail de contacte:* marc.lopezse@e-campus.uab.cat
- *Menció realitzada:* Enginyeria del Software.
- *Treball tutoritzat per:* Josep Lladós Canet (departament de Ciències de la Computació)
- *Curs 2018/19*
- Secció 2, es descriuen els objectius principals i secundaris del projecte.

- Secció 3, es defineix l'estat de l'art.
- Secció 4, representa la planificació seguida durant la realització del projecte.
- Secció 5, s'explica la metodologia de treball realitzada.
- Secció 6, es presenten els resultats i proves realitzades i obtingudes.
- Secció 7, s'expliquen les conclusions finals extretes i futur treball.

2 OBJECTIUS

2.1 Objectiu principal

L'objectiu principal del projecte és desenvolupar una eina de visualització d'imatges a partir d'una base de dades d'aquesta modalitat, aplicant algorismes de visualització BigData [2]. Es vol que amb aquest programa faciliti de forma àgil la cerca d'imatges, depenent de les necessitats que necessiti l'usuari, ja sigui pel contingut de la imatge o similituds entre totes les de la base de dades. Aquest aplicatiu, seria molt eficient per aquelles bases de dades que disposen d'un volum ocupat molt gran.

Com a objectius clau que tenim per aquest projecte són:

- Estudi d'eines de cerca i visualització existents.
- Creació d'una aplicació utilitzant un procés de desenvolupament de software
- Incorporació d'algorismes que utilitzin criteris de semblança entre imatges
- Integrar algorismes de cerca i visualització extrets de l'àmbit BigData.

2.2 Objectiu secundari

Un objectiu més avançat seria utilitzar l'aplicatiu en diferents perspectives u objectes d'altres modalitats, que serien:

- L'ús de diferents característiques d'imatges (data de creació, tags associats, colors predominants, etc.)
- Imatges en format GIF
- Diferents formats de vídeo
- Documents PDF

3 ESTAT DE L'ART

Durant la realització del projecte, s'ha estat investigant els possibles algorismes a aplicar en el projecte (PCA [9] i T-SNE). Donat els objectius del projecte i la gran varietat de requisits que es poden donar per a futures funcions a implementar, era més òptim la utilització de T-SNE.

El t-SNE (T-distributed Stochastic Neighbor Embedding) és un algorisme de l'àmbit de Machine Learning per a la visualització de les dades. Tracta de d'una tècnica de reducció de dimensionalitat no lineal, adequada per la

incorporació d'un gran volum de dades en un espai de dimensió més baixa (de 2 o 3 dimensions).

La seva funcionalitat és, mitjançant un gràfic, representa el conjunt de dades a mostrar i classifica cada dada depenent de la semblança que hi ha entre elles. Es a dir, les dades que tenen més similituds entre elles estaran més a prop i les que tenen menys llavors estaran més lluny. La característica de semblança depèn del criteri que es vol avaluar (contingut d'imatges, dates de publicació, pacients amb enfermetats similars, etc).

Com a referència de funcionalitat en la cerca d'imatges, s'ha fet una investigació dels motors de cerca d'imatges disponibles, per tal d'agafar idees i adaptar-ho a algorismes de bigdata. Els motors de cerca analitzats han sigut: Google Images, Bing, Yahoo i DuckDuckGo.

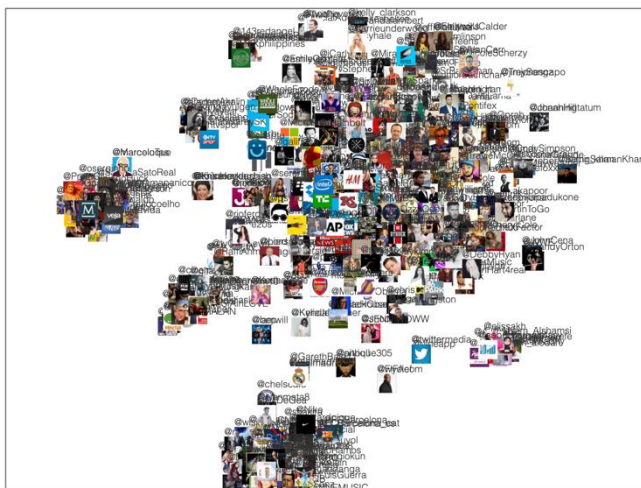


Fig.1: T-SNE aplicat amb tweets de diferents usuaris[11].

Durant la investigació de l'algorisme, s'ha buscat aplicacions similars als objectius del projecte però en diferents àmbits. Les aplicacions trobades han sigut l'aplicació de l'algorisme T-SNE per a la classificació de Tweets depenen del contingut del missatge [Figura 1], classificació de similituds de paraules depenents de la longitud i el continguts dels caràcters d'aquesta o fins i tot de detecció de números en una imatge [3].

4 PLANIFICACIÓ

Per definir la planificació s'ha estudiat al detall de quines tasques a realitzar per poder fer el projecte. Seguint els estàndards de desenvolupament del software, es divideix en 4 fases principals (especificacions, disseny, implementació i proves) i després uns informes d'entrega del projecte cada vegada que es tenia una versió finalitzada.

La planificació proposada per la realització del projecte és la següent:

- Estudi de l'algorisme t-SNE i documentació d'aplicatius similars

- Fase de Disseny:
 - Disseny preliminar de l'aplicatiu, tenint en compte els objectius i les necessitats de l'usuari
 - Elaboració i lliurament d'informe de progrés I
 - Reunió per discutir i analitzar el disseny preliminar
 - Disseny final de l'aplicatiu
- Fase d'implementació:
 - Implementació basica front-end de l'aplicatiu
 - Reunió per evaluar el front-end i debat sobre back-end
 - Incorporació back-end amb l'algorisme t-SNE
 - Elaboració i lliurament d'informe de progrés II
 - Implementació front-end amb resultats de l'algorisme
 - Reunió sobre el testeig de l'aplicatiu
- Testing de l'aplicatiu amb un conjunt de proves
- Elaboració i lliurament d'informe final
- Reunió final del projecte

Això es degut a que a l'aplicatiu web es poden fer una sèrie de modificacions mentre s'inserta l'algorisme, es a dir, el disseny pot canviar a mesura que avança el projecte.

De cara als processos planificats, s'han presentat una sèrie de dificultats de cara a escollir el millor llenguatge de programació per a l'algorisme, que ha trigat més temps del previst. Es tracta d'escollir l'algorisme t-SNE aplicat entre Python i Javascript.

Durant una àrdua investigació i proves, l'aplicatiu s'ha desenvolupant en els llenguatges PHP, HTML, CSS i Javascript. Per la estructura d'aquesta, s'ha aplicat l'arquitectura Model Vista Controlador (MVC) degut a l'organització més eficient entre les parts front-end i back-end, per tal d'optimitzar els processos. Per evitar dependències d'un servidor extern durant el desenvolupament, s'ha utilitzat XAMPP (veure l'apartat 5.2), el qual permet la compilació del codi PHP sense requerir d'internet.

El procés de la planificació relacionada amb la implementació bàsica front-end i la incorporació back-end de l'algorisme T-SNE ha donat una sèrie de dubtes a l'hora d'utilitzar un llenguatge o un altre (com s'ha mencionat anteriorment). Al final s'ha arribat a la conclusió que per càlculs matemàtics i extracció d'informació és millor l'ús de Python i fer servir Javascript per representar els resultats de forma dinàmica.

5 METODOLOGIA

Avaluant de molt a prop les tasques planificades a realitzar, al final s'opta per la utilització de la metodologia de treball de forma iterativa [Figura 2].

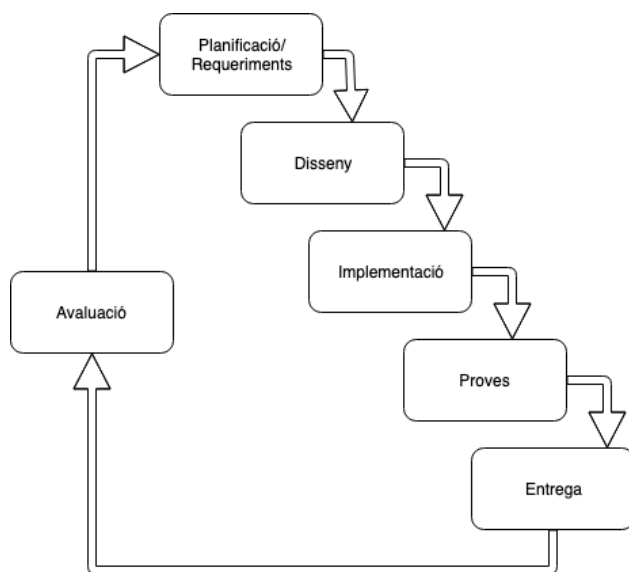


Fig.2: Cicle de vida iteratiu del projecte [8].

5.1 Requeriments trobats

Per poder realitzar el projecte, van sorgir una sèrie de requeriments necessaris per poder satisfer els principals objectius, els quals es classifiquen segons en requeriments funcionals i no funcionals.

5.1 Requeriments funcionals

- El sistema ha de permetre accedir de forma senzilla al gràfic.
- Al seleccionar una imatge, el sistema ha de permetre accedir al tamany original de la imatge.
- Al passar per sobre d'una imatge, el sistema permetrà veure una previsualització de la imatge que apunta el mouse.
- El gràfic mostrarà totes les imatges amb miniatures.
- El gràfic ha de mostrar una gran quantitat d'imatges per tal de veure l'efectivitat de l'algorisme.
- El sistema mostrarà una previsualització dels resultats de l'algorisme.
- La plataforma web pot ser utilitzada en qualsevol sistema operatiu.
- L'aplicatiu web s'ha de poder utilitzar en el navegador web Chrome, Firefox i Safari.

5.2 Requeriments no funcionals

- El sistema haurà de ser capaç de processar com a mínim 1.000 imatges amb l'ús de l'algorisme t-SNE.
- El sistema ha de ser capaç d'interpretar les dades i l'arquitectura del software plantejada en un navegador web.
- L'arquitectura de la plataforma web ha de seguir el sistema Model Vista Controlador (MVC).
- El sistema ha de ser capaç de captar com a mínim 3 events amb el mouse [10] per poder fer diferents interaccions.
- L'interfície gràfica ha de ser representada amb codi HTML, CSS i Javascript.

5.2 Eines utilitzades

Per la realització del projecte s'han utilitzat diferents llenguatges de programació i formats de fitxer, el qual són:

- **Python**, per l'utilització de l'algorisme t-SNE i la preparació de les dades per aquest.
- **JSON**, per exportar les dades obtingudes de l'algorisme.
- **PHP**, per controlar les funcionalitats i construcció de l'arquitectura de l'aplicatiu.
- **Javascript**, per representar les dades del fitxer JSON i mostrar-les en un gràfic.
- **HTML** per la estructura de l'aplicatiu.
- **CSS**, pel disseny front-end de la plataforma web.

Pel que fa les eines, s'han utilitzat diferents aplicacions:

- **Spyder**: és un entorn de desenvolupament interactiu utilitzat per al llenguatge python.
- **PHPstorm**: és un entorn de desenvolupament interactiu que permet construir l'arquitectura de l'aplicatiu web a base del llenguatges de programació HTML, PHP i CSS (entre d'altres).
- **XAMPP**: és un paquet de programari lliure que conté principalment el sistema de gestió de bases de dades MySQL, un servidor web Apache i els intèrprets pels llenguatges PHP i Perl. El seu nom es un acrònim el qual significa: X(qualsevol sistema operatiu), A(Apache), M(MariaDB/MySQL), P (PHP), P(Perl). A partir de la versió 5.6.15, XAMPP canvia la seva base de dades MySQL per MariaDB.
- **Google Chrome**: navegador web de software primitiu desenvolupat per Google. S'ha utilitzat per fer les diferents proves de cara a la visualització de la part front-end i els events que s'implementen en back-end.

5.3 Disseny del projecte

En la fase de disseny del projecte, van sorgir uns dubtes a l'hora de l'utilització de l'algorisme t-SNE i amb quin

llenguatge de programació s'utilitzarà. Després haver escollit, es busca una manera de comunicar el codi python amb l'aplicatiu web. La solució va ser utilitzar JSON, el qual externalitzem les dades desde Python i el pot interpretar el javascript de l'aplicatiu.

Una vegada decidit les comunicacions dels mòduls, la arquitectura del software plantejada per aquest projecte és la representació gràfica que es pot veure a la següent imatge [Figura 3].

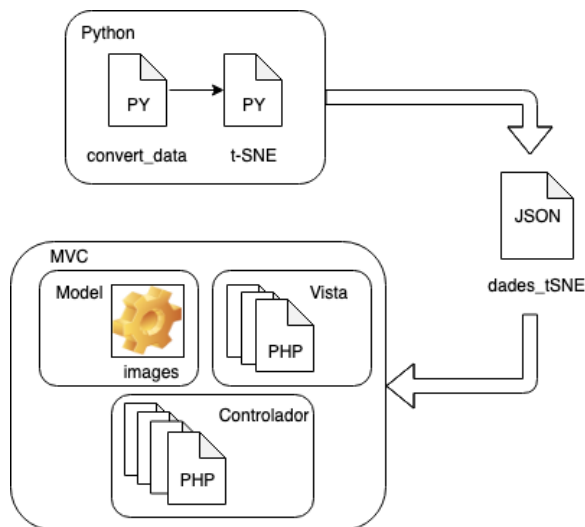


Fig.3: Arquitectura del projecte.

En l'arquitectura podem veure 2 blocs:

- Bloc de python, conté la preparació de les dades i el propi algorisme.
- Bloc Model Vista Controlador, que conté tota l'arquitectura de l'aplicatiu web. Aquest bloc es subdivideix en altres 3 blocs, per separar la part front-end (vista), back-end (controlador) i la part de consultes (model).

L'arxiu JSON compleix un paper important, ja que és el que conté tots els resultats extrets de l'algorisme. L'utilitza la part javascript de l'aplicatiu per donar forma a les dades i representar-ho en un gràfic.

5.4 Implementació

Es va haver de programar i adaptar una sèrie de models específics per tal de poder construir l'aplicatiu correctament.

El desenvolupament que es va fer, per la part d'algorítmica, és la següent:

- Adaptació de les dades (convert_data.py). L'algorisme en si requereix d'una estructura de dades per tal de poder-les processar. S'agafen una determinada quantitat d'imatges a processar i després s'externalitzen en un fitxer a part, que després el propi algorisme farà servir. S'ha de tenir en compte que el procés

de conversió d'imatges pot trigar molt, dependent de la quantitat d'imatges que s'agafi.

- Execució de l'algorisme t-SNE (TSNE.py). S'agafen les dades comprimides anteriorment i s'executa l'algorisme. Els resultats que dona, es representen en un graph amb matplotlib, per tal de comprovar que funciona correctament. Finalment, les dades resultants, s'exporten en un fitxer JSON[4].

Per el que es refereix a la part de l'aplicatiu es van crear moltes parts, les quals es referenciarán les més importants. Aquestes parts són:

- Index.php. Aquí es vinculen tots els controladors de l'aplicatiu. Des d'aquest fitxer, s'accedeix a gairebé qualsevol secció de l'aplicatiu.
- Controller. Aquesta part de l'aplicatiu s'encarrega de la part back-end i les crides de les parts front-end.
- View. És la secció on estan totes les parts front-end de l'aplicatiu. Des d'aquí s'encarrega que l'usuari pugui rebre la informació desitjada amb un sistema user-friendly.
- grafico-tsne.js. El codi javascript que construeix el gràfic on es mostren totes les imatges. Amb la informació de l'arxiu JSON, es construeix una estructura que agafa i crea els tags per la representació de cada imatge. També es controlen els events amb el mouse, de tal manera que podem veure una imatge qualsevol del grafic de forma ampliada.

Durant la implementació del gràfic, es va trobar un problema que provocava que algunes imatges no es poguessin visualitzar degut al solapament d'imatges i que els punts inicials de cada una eren molt petits. Llavors es va buscar una solució adequada per tal de facilitar l'accés de les fotografies de manera user friendly. La resposta van ser els diagrames de Voronoi [Figura 4], el qual va facilitar molt el treball de desenvolupament i permet seleccionar qualsevol imatge per les regions que es creen.

Per entendre més el concepte, el diagrama de Voronoi consisteix en una estructura senzilla e intuïtiva formada per una divisió de regions d'un conjunts de punts en un pla [6].

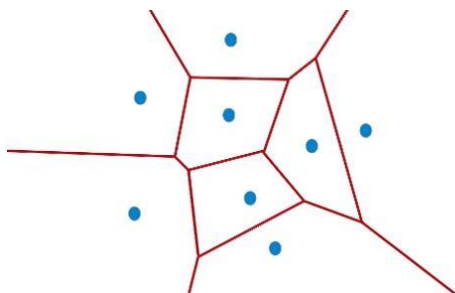


Fig.4: Exemple d'un diagrama de Voronoi

El que fa es dividir l'espai en tantes regions com punts/objetes es disposen, de tal manera que cada punt li és assignat una regió formada per tot l'espai que està més a prop d'ell.

Per veure el gràfic de l'aplicatiu de forma general, les imatges s'han de veure en miniatura. Per tal de poder seleccionar una imatge, creem una regió per cada una i es delimiten les regions per proximitat entre els punts més propers a cadascun. Això ens facilita molt l'accés a la visualització completa de les imatges.

6 RESULTATS

Per tal de poder explicar els resultats, s'han classificat dependent del tipus de llenguatge utilitzat i també dependent de quina part de l'aplicatiu forma part. Es classifiquen els resultats en:

- Algorisme T-SNE amb implementació en python.
- Dades extretes referents a les imatges a representar, en format JSON.
- Representació gràfica dels resultats amb l'ús de d3js.

6.1 Algorismica amb Python

Després de l'adaptació de l'algorisme, al final s'han fet una sèrie de proves per poder provar com reacciona i quines falles es poden trobar. S'ha realitzat l'anàlisi a partir d'utilitzar primerament unes 200 imatges a l'atzar [Figura 5] i, seguidament 1.000 imatges.

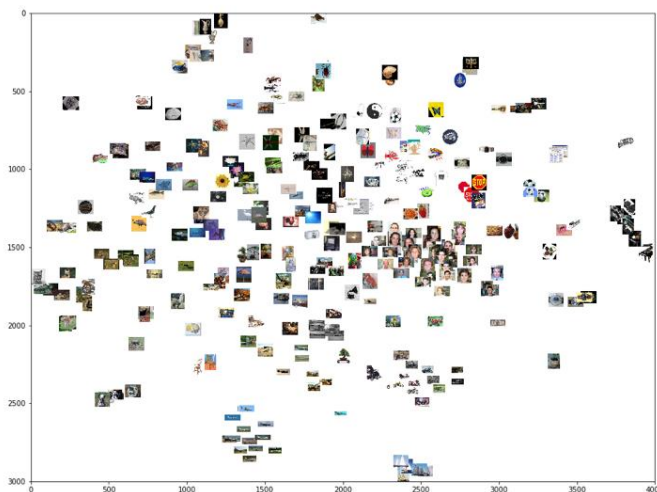


Fig.5: Gràfic de classificació T-SNE amb 200 imatges.

Analitzant el resultat, podem observar imatges una mica disperses, donant una sensació de que hi ha poques similituds i, per tant, no es pot comprovar correctament, ja que es requereix de moltes més imatges. Per tant, al veure que es necessiten més, s'incrementa la quantitat

d'imatges a analitzar fins a 1000 [Figura 6] (per simular una gran quantitat d'informació).

En aquest cas, podem observar grans acumulacions d'imatges, el qual mostren les similituds que tenen i, per tant, veure el seu potencial. Podem veure aquestes similituds mitjançant petits núvols o agrupacions d'imatges que es formen al comparar distàncies entre imatges que tenen uns valors similars.

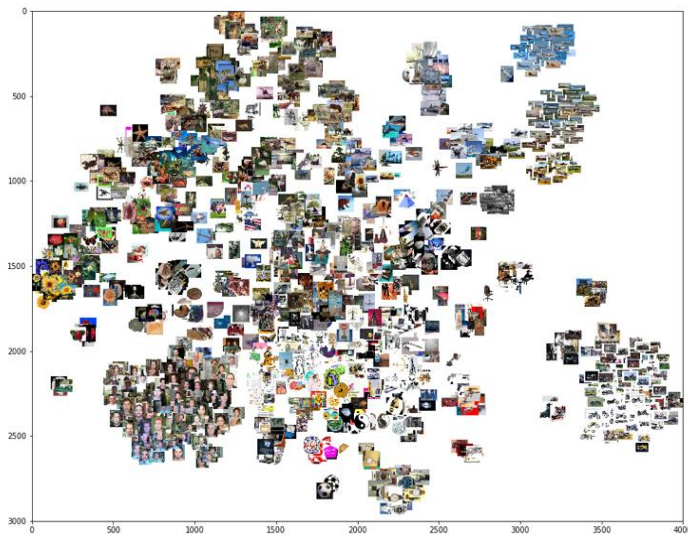


Fig.6: Gràfic de classificació T-SNE amb 1.000 imatges.

Com a conclusions d'aquests resultats, l'algorisme mostra molta eficiència quantes més dades analitza.

6.2 Extracció de dades en format JSON

Al tenir els resultats de l'execució de l'algorisme, es requereix una adaptació de les dades per tal de poder-les representar, mitjançant l'ús de javascript. Per aquest motiu, es va optar per externalitzar les dades en format JSON. El motiu d'aquesta decisió és degut a que la complexitat de traducció i comunicació és complicada de forma directe entre PHP i Python. Amb el format JSON, podem donar forma desde Python, externalitzar-ho i després fer una lectura amb Javascript.

El format que s'ha plantejat per l'externalització de les dades extretes de l'algorisme és el següent:

- X: posició "x" del gràfic extret.
- Y: posició "y" del gràfic extret.
- Image: directori on està localitzada la imatge.

Es crea un array, el qual conté per cada imatge les característiques anteriors. Cada posició de l'array conté la informació d'una imatge específica, escollida per l'algorisme. El tamany de l'array equivaldria a la quantitat d'imatges que es volen representar en el gràfic.

Amb aquest mètode d'extracció, la complexitat de representació es redueix, ja que l'execució de l'algorisme es

fa apart de l'aplicatiu i facilita la representació de les imatges en un gràfic.

6.3 Representació amb D3.js

Per tal de poder representar en l'aplicatiu web les dades extretes en el format JSON, s'ha utilitzat la llibreria d3.js [5]. Aquesta llibreria permet fer una representació gràfica més dinàmica de les dades, amb la utilització del HTML, CSS i SVG.

L'utilització d'aquesta llibreria suposa un repte, ja que permet crear nous tags html de forma dinàmica i sense control d'aquesta pot provocar un caos entre el contingut de l'aplicatiu i les dades a representar. Per tant es planteja representar una forma geomètrica bàsica per la pràctica i control d'aquesta llibreria [7].

6.3.1 Representació bàsica d'aprenentatge

Es van representar formes geomètriques per simular que s'introdueix una imatge en un punt a l'espai en el gràfic per entendre les diferents opcions que ofereix la llibreria [Figura 7]. Mitjançant representacions i les posicions de les figures, es podia permetre obtenir una idea bàsica de com plantejar el disseny del gràfic.

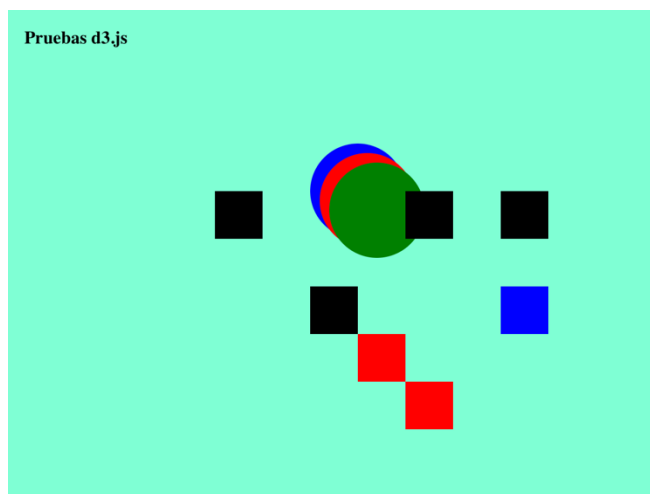


Fig.7: Representacions bàsiques amb d3.js

Una vegada entesos els conceptes principals, es programa la part principal de la representació, mitjançant la classificació i creació de diverses funcions, que permeten el correcte funcionament del gràfic.

Primer de tot es van crear les principals propietats del gràfic i les representacions dels eixos "x" e "y". Després es van representar el tamany de les miniatures de les imatges i, per últim els events relacionats amb el mouse.

Després de fer les primeres representacions, va sorgir el problema de que era molt complicat seleccionar una imatge específica degut al seu tamany i el solapament d'imatges. Per tal de que no quedin espais buits i sense poder fer interacció amb les imatges, s'aplica el diagrama

de Voronoi per accedir-hi a aquestes de forma més senzilla.

6.3.2 Resultats finals.

Al completar l'estructura dedicada a la representació de les imatges, va sorgir un problema que impossibilitava l'accés a moltes imatges. Això va ser degut a que les imatges es representaven a escala real i no es podia percebre la sensació de tenir un gràfic representat [Figura 8].

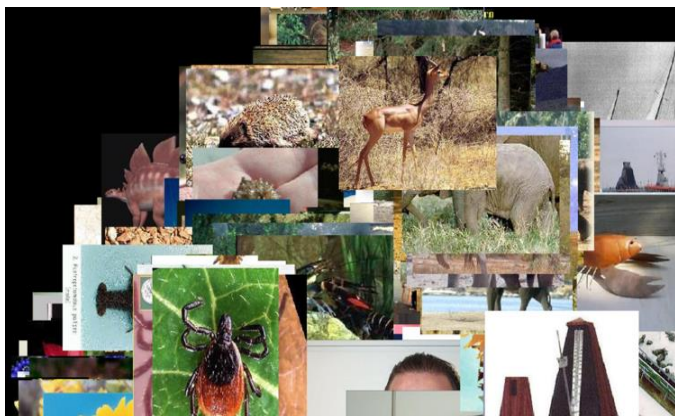


Fig.8: Primers resultats de la representació del gràfic.

Es van fer una sèrie de modificacions per solucionar aquest problema. Una d'aquestes modificacions era aplicar les regions de Voronoi per tal de poder seleccionar una imatge més fàcilment, degut a la gran quantitat d'imatges. No es podia fer que una imatge sencera fós seleccionable, ja que al darrere d'aquesta hi podien haver més. Llavors mitjançant els punts de representació de cada imatge (x, y) i delimitar regions, suposa una solució més user-friendly. A més, s'incorpora una funcionalitat que permet previsualitzar la imatge en la qual apunta el mouse [Figura 9].



Fig.9: Funcionalitat de previsualització d'imatge.

Una vegada haver fet les correccions necessàries, es

pot observar en la [Figura 10] com queda al final la representació del gràfic. Es pot veure de forma general totes les imatges dependent del contingut de cadascuna. S'han provat varies opcions per canviar el background, però al final es va observar que el color negre mostrava les imatges de forma més clara i no forçava la vista a l'usuari.



Fig.10: Representació final en petit de tot el gràfic, dins de l'aplicatiu.

7 CONCLUSIÓ I TREBALL FUTUR

7.1 Conclusions

En aquest treball s'han estudiat diferents algorismes de visualització de dades per a navegació en col·leccions de fotografies. En particular, s'ha desenvolupat una aplicació web que integra l'algorisme t-SNE, amb la funcionalitat de representació de similituds entre les diferents imatges, representades en un gràfic.

Durant la realització del projecte, s'ha vist que per una utilització òptima de l'algorisme, requereix una gran quantitat d'informació (en aquest cas d'imatges), ja que és quan es veu el potencial que ofereix aquest. Analitzant l'algorisme hi ha un inconvenient, que fa referència a les dades a preparar per poder-les processar en l'algorisme. A major quantitat de dades per representar a l'algorisme, més efectivitat d'anàlisi de comparació d'imatges però també augmenta el temps per convertir i prepara les imatges també creix.

Pel que fa amb els llenguatges de programació utilitzats, he pogut deduir que si un d'ells ho domines, utilitzar-lo i buscar una solució de com comunicar les parts interessades. Extreure aquesta deducció va ser una mica difícil, degut a que es van buscar algorismes directament amb javascript, però el coneixement d'aquest llenguatge que disponia era bàsic. Analitzar tot el codi era molt confós i molt llarg. Llavors vaig pensar en utilitzar Python que era el que més he utilitzat i a la vegada és molt recomanable per programar càlculs matemàtics i execució d'algorismes estadístics.

Gràcies als coneixements extrets durant la carrera i l'experiència laboral obtinguda, he pogut analitzar amb

més detall els problemes que han sorgit i com solucionar-los de manera efectiva. A base de tutorials que es disposen en internet i el coneixement propi, he pogut aprendre i millorar en les àrees de coneixement que anava una mica perdut.

Durant la realització del projecte, sempre hi havia problemes de construcció del gràfic i de la pròpia plataforma. Aquests problemes m'han ajudat cada vegada més a entendre com organitzar les estructures de l'aplicatiu o modificar-les de tal manera que trobava una solució senzilla per una implementació complexa (analitzar des de diferents perspectives).

7.1 Treball futur

Aquest projecte, es pot expandir afegint nous mòduls i noves funcionalitats a l'algorisme, de tal manera que en comptes d'agrupar les imatges per la semblança pròpia entre elles, es podrien posar nous filtres (per exemple: classificar per data de creació, per etiquetes, per tamany que ocupen, etc). A més, també es pot expandir de tal manera que en comptes d'analitzar imatges, es puguin classificar GIFs, vídeos, documents PDF u altres tipus de fitxer.

També cal dir que, en comptes de contenir directament les imatges dins del propi projecte (que actuarà com a part del model), es podrien insertar dins d'una base de dades, amb les característiques resultants extretes de l'algorisme t-SNE. Les modificacions que s'aplicarien serien que, en comptes de manipular un JSON directament amb javascript, es farien consultes a la base de dades per extreure la informació rellevant.

Aquest projecte es pot expandir de diferents maneres, depenent dels objectius que es volen assolir. Un detall important a tenir en compte es que eficiència de l'algorisme es pot veure amb l'introducció d'una gran quantitat de dades.

Agraïments

M'agradaria agrair en primer lloc al meu tutor, Josep Lladós Canet, per ajudar-me a entendre diferents conceptes i guiarme durant aquest procés del projecte. Els seus consells m'han ajudat a enfocar el projecte d'una manera que diferent al que normalment plantejo.

Agraeixo a les persones que han realitzat un estudi sobre l'algorisme T-SNE per ajudar-me a entendre com funciona i sobre els diferents conceptes que s'han necessitat per la realització d'aquest projecte (llibreria d3.js, diagrames de Voronoi).

També als meus amics per donar la seva opinió respecte als desenvolupaments fets, de cara a la part front-end de l'aplicació, ja que gràcies a això l'he pogut millorar i he pogut distribuir la informació de cara a que sigui més atractiva i user-friendly.

Per últim vull agrair als meus pares per tot el suport que m'han estat donant durant la realització del projecte

BIBLIOGRAFIA

- [1] Rodríguez-Vaamonde, Sergio; Torre-Bastida, Ana-Isabel; Garrote, Estibaliz. "Tecnologías big data para análisis y recuperación de imágenes web". El profesional de la información [en línia]. 2014, vol. 23, n. 6, noviembre-diciembre, p. 567-574. [Disponible a internet: <http://www.elprofesionaldelainformacion.com/contenidos/2014/nov/02.pdf>]
- [2] Wikipedia, "T-distributed Stochastic Neighbor Embedding" [en línia]. Disponible a internet: https://en.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding
- [3] Cyrille Rossant. An illustrated introduction to the t-SNE algorithm. Oreilly [en línia]. 3 de març de 2015. Disponible a internet: <https://www.oreilly.com/learning/an-illustrated-introduction-to-the-t-sne-algorithm>
- [4] Daniel Rodríguez, "Archivos JSON on Python: lectura y escritura". Analytics Lane [en línia]. 16 de juliol de 2018. Disponible a internet: <https://www.analyticslane.com/2018/07/16/archivos-json-con-python/>
- [5] Marco Godínez, "Introducción a D3.js". Javascriptmx [en línia]. Disponible a internet: <http://javascriptmx.com/blog/introduccion-a-d3js/>
- [6] Clara Grima, "Cada uno en su región y Voronoi en la de todos". Naukas [en línia]. 23 de desembre de 2011. Disponible a internet: <https://naukas.com/2011/12/23/cada-uno-en-su-region-y-voronoi-en-la-de-todos/>
- [7] d3Vienno. "D3.js tutorial - 3 - Basic SVG shapes". Youtube.[en línia].Publicat el 13 de març de 2013. < <https://www.youtube.com/watch?v=TR39nfAW1dw> >
- [8] Jorge Saiz. "Ciclo de vida iterativo. Los proyectos con procesos cíclicos"[en línia]. 27 desembre de 2018. Disponible a internet: <https://jorgesais.com/blog/ciclo-de-vida-iterativo/>
- [9] Cristina Gil Martínez, "Maquinas de vector soporte". Rpubs [en línia]. Juny de 2018. Disponible a internet: https://rpubs.com/Cristina_Gil/PCA
- [10] Anthony Nosek, "D3.js Mouse Events". Stator [en línia]. Disponible a internet: <http://www.stator-afm.com/tutorial/d3-js-mouse-events/>
- [11] Andrej Karpathy, "Visualising Top Tweeps with t-SNE, in Javascript". Andrej Karpathy blog [en línia]. 2 de juliol de 2014. Disponible a internet: <http://karpathy.github.io/2014/07/02/visualizing-top-tweeps-with-t-sne-in-javascript/>