

APLICACIÓ TRADUCTORA DE LLENGUATGE DE SIGNES

Ferran de Gea Rodríguez
2022-23

INS Vilatzara

Tutor: Jaume Linares
Grup: 2BTX B
Matèria: TR (Tecnològic)

ABSTRACT EN ANGLÈS

Today's society relies on communication more than ever. However, a problem arises when trying to communicate with a hearing-impaired person, if more than simple messages want to be conveyed, both parties need to understand sign language, which is obviously not a viable short-term solution. That ultimately leads to hacky and impractical solutions like writing on paper or a note-taking app.

This paper presents the early stages of a solution to the previously mentioned problem, automated sign language detection and transcription wrapped in a user-friendly smartphone application. While not excessively resource-intensive, it allows for quick deployment and usage, reducing the time of confusion between the two involved in the communication attempt.

The approach explained in this paper contains a very primitive implementation of a technology that could eventually be used to provide real-time translation of all types of signs. It consists of the detection of the most important features of the hand and its classification using a hand-craft database. Currently, only one-handed static signs consisting of individual letters are supported, nevertheless, if re-structured, it can be extended to allow for two-handed non-static signs.

The contents of the document include an in-depth analysis of the Spanish sign language, ranging from its history to its structure; a superficial explanation of neural networks, one of which is used in the detection technique; a complete description of the technique used, which uses two methods for the detection and classification of the sign, and its implementation on a web environment; an overview of the design and programming of the application, and a showcase of the results obtained.

ABSTRACT EN CASTELLÀ

Nuestra sociedad actual depende más que nunca de la comunicación entre nosotros. Sin embargo, cuando se intenta comunicarse con una persona con discapacidad auditiva surge un problema: si se quiere transmitir algo más que mensajes sencillos, ambas partes deben entender el lenguaje de signos, lo que obviamente no es una solución viable a corto plazo. Esto acaba llevando a soluciones poco prácticas como escribir en papel o en una aplicación de notas.

Este artículo presenta las primeras fases de una solución al problema mencionado anteriormente, la detección y transcripción automatizada de la lengua de signos integrada en una aplicación móvil fácil de usar. Que al no requerir demasiados recursos, permite un despliegue y uso rápidos, reduciendo el tiempo de confusión entre los dos implicados en el intento de comunicación.

La propuesta explicada en el documento trata de una implementación muy primitiva de una tecnología que podría utilizarse eventualmente para la traducción en tiempo real de todo tipo de signos. Esta consiste de la detección de los rasgos más prominentes de la mano y su posterior clasificación teniendo en cuenta una base de datos creada manualmente. Actualmente, solo soporta signos estáticos de una mano que consisten en letras individuales, sin embargo, si se reestructura, puede ampliarse para admitir signos en movimiento de ambas manos.

Los contenidos de este documento incluyen un análisis en profundidad de la lengua de signos española, desde su historia a su estructura y gramática; una explicación superficial del funcionamiento de las redes neuronales, usadas en la técnica de detección; una descripción en detalle de la técnica usada, la cual usa dos métodos diferentes para su detección y clasificación de los signos, y una implementación de esta en un entorno web; una explicación del proceso de diseño y creación de la aplicación, y una exposición de los resultados finales.

ÍNDEX

ABSTRACT EN ANGLÈS	1
ABSTRACT EN CASTELLÀ	2
ÍNDEX	3
1. INTRODUCCIÓ	4
1.1 OBJECTIUS	5
1.2 MÈTODE DE TREBALL	6
1.3 PUNT DE PARTIDA	6
2. LA LLENGUA DE SIGNES	7
2.1 QUÈ ÉS?	7
2.2 NOMENCLATURA	7
2.3 HISTÒRIA DE LA LSE	8
2.4 ESTRUCTURA	9
2.5 ALFABET DACTIOLÒGIC	13
2.6 ÚS EN EL PROJECTE	14
3. PLANIFICACIÓ	15
3.1 PLA DEL PROTOTIP	15
3.2 PLA DE L'APLICACIÓ FINAL	16
4. LA INTEL·LIGÈNCIA ARTIFICIAL	18
4.1 QUÈ ÉS UNA XARXA NEURONAL?	18
4.2 TIPUS DE XARXES	20
5. DETECCIÓ	25
5.1 POSICIÓ DELS DITS	25
5.2 CLASSIFICACIÓ	27
5.2.1 EL TEOREMA DE BAYES	27
5.2.2 CLASSIFICADOR DE BAYES INGENU	30
5.3 LIMITACIONS	38
6. PROTOTIP	39
7. APLICACIÓ	40
7.1 FRONTEND	40
7.2 BACKEND	43
8. CONCLUSIONS	46
9. ANNEX	48
10. FONTS DE CONSULTA	48

1. INTRODUCCIÓ

En la societat actual és molt important poder comunicar-se clarament amb tot tipus de persones, però això no és sempre possible. Normalment resulta difícil comunicar-se amb persones sordes o amb problemes auditius, la majoria de discapacitats poden comunicar-se entre ells, degut a que tenen un idioma comú, el llenguatge de signes. Nogensmenys, la majoria de la població sense discapacitat auditiva no pot entendre'l.

Aquest article proposa una solució, la traducció automàtica del llenguatge de signes amb una aplicació mòbil. Sent més eficaç que l'escriptura en una llibreta o una aplicació de notes per poder comunicar-se, aquesta proposta permet el llançament ràpid d'una aplicació que pot traduir els signes en temps real. A l'enfocar la càmera de darrere el mòbil al parlant, l'aplicació detectarà la posició de les mans i realitzarà la traducció. El resultat es mostrerà en pantalla en un quadre de text.

Degut a limitacions de temps i mà d'obra, només s'intentarà crear una versió primitiva de l'aplicació, aquesta no incorporarà tots els matisos del llenguatge de signes i podrà ser descrita més aviat com a una demostració tècnica que es podria millorar en el futur, que una aplicació funcional usable. No obstant, sempre es tindrà en compte l'aspecte comercial en la seva realització.

1.1 OBJECTIUS

L'objectiu principal d'aquest treball és crear una aplicació que permeti detectar i interpretar el llenguatge de signes bàsic utilitzant només un telèfon mòbil. Durant el procés de realització d'aquesta aplicació es buscarà assolir els següents punts:

1. Crear un sistema d'identificació dels trets més rellevants que fan la forma de la mà:
 - Recerca de mètodes d'identificació de trets rellevants de les mans.
 - Creació d'un programa provisional funcional només en un ordinador.
2. Crear un sistema que classifiqui aquests punts segons una base de dades amb tots els signes:
 - Recerca de mètodes de classificació de dades numèriques.
 - Creació d'un programa provisional funcional només en un ordinador.
 - Integració del sistema amb l'anterior.
3. Dissenyar una aplicació mòbil i incorporar els anteriors sistemes tenint en compte el seu rendiment en dispositius de baixa gamma:
 - Convertir els sistemes prèviament només funcionals en un ordinador a un llenguatge de programació que es pugui executar en una aplicació mòbil.
 - Disseny visual de la aplicació.
 - Creació de les pantalles de l'aplicació funcionals.
 - Integració del reconeixement de signes a l'aplicació.

Durant la realització dels anteriors passos sempre es tindrà en compte la finalitat comercial i pública de l'aplicació.

1.2 MÈTODE DE TREBALL

El mètode de treball que s'utilitzarà per assolir els objectius anteriors consisteix en els següents passos:

1. Recerca d'informació (si és necessària).
2. Planificació i estructuració de les accions necessàries per completar la tasca.
3. Realització i experimentació.
4. Avaluació del resultat, si no s'està satisfet es tornarà a realitzar algun dels passos anteriors.

1.3 PUNT DE PARTIDA

Vàries empreses i individus han fet programes similars. La majoria estan en desenvolupament, en fase de proves, només funcionen en un ordinador o són de pagament. Uns exemples d'empreses que ofereixen serveis d'interpretació, en el moment que s'està escrivint aquest article, són Signall (de pagament) i Slait (en fase de proves).

2. LA LLENGUA DE SIGNES

2.1 QUÈ ÉS?

Utilitzada majoritàriament per persones amb problemes auditius o completament sordes, la llengua de signes és un idioma de caràcter visual, gestual i espacial amb una gramàtica pròpia que reuneix totes les característiques i compleix les mateixes funcions que qualsevol altra llengua oral. Tanmateix, no existeix una llengua internacional, cada regió té la seva pròpria versió en la qual poden variar molts aspectes, des dels més obvis com la forma dels signes fins a l'estructura de l'oració. A Espanya existeixen dues llengües de signes, l'espanyola i la catalana, comptant amb més de 70.000 parlants en total entre ambdues, dels quals 25.000 usen la catalana.

Degut al seu gran nombre de parlants, en aquest projecte s'usarà sempre la llengua de signes espanyola (LSE).

En la LSE s'usa l'expressivitat de les mans, la cara i el cos. Aquesta posseeix la seva pròpria estructura i permet a les persones sordes utilitzar-la com a principal via de comunicació, sense necessitat de saber cap llengua oral.

2.2 NOMENCLATURA

Per entrar més en detall sense allargar massa les explicacions utilitzaré algunes abreviacions, acrònims i sigles:

- **LSE** = Llengua de signes espanyola
- **LSC** = Llengua de signes catalana
- **Signar** = Realitzar un signe

Totes les paraules representant un signe estaran en majúscula.

2.3 HISTÒRIA DE LA LSE

A Espanya, la història de la llengua de signes data des del segle **XVI**. En els monestirs d'aquesta època es van començar a fer ús signes per poder comunicar missatges importants respectant l'obligació de guardar silenci. A més a més, els capellans també intentaven educar als nens sords o amb problemes auditius. **Pedro Ponce de León** va ser el primer capellà en adonar-se de la necessitat d'aquests nens i va començar a emprar el seu sistema gestual de comunicació.

En el segle **XVII**, **Manuel Ramírez de Carrión**, un autor i educador, es va guanyar el títol del precursor de la pedagogia de sordmuts, ensenyant-los la llengua de signes de l'època i a llegir i escriure en espanyol.

No obstant, no va ser fins a la **meitat del segle XVII** quan es va publicar un llibre anomenat *Escuela española de sordomudos o arte para enseñarles a escribir y hablar el idioma español* escrit per **Lorenzo Hervás y Panduro**. On està escrit tot el necessari perquè els sordmuts puguin integrar-se a la societat, sabent escriure i podent comunicar-se.

El pas més gran a l'estandardització d'aquesta llengua es dona amb la publicació del *Diccionario de mimica y dactilología* de **Francisco Villabrille** al segle **XIX**. On s'inclouen més de 1500 signes descrits en detall.

Durant aquest mateix segle **s'inauguren nombroses escoles** per a sordmuts i cecs. Allà es desenvolupa la interacció lingüística i social en àmbits de la vida quotidiana. Aquestes escoles i els seus alumnes donen lloc a l'inici de les protollengües de signes **espanyola i catalana**, ja més semblants a les actuals.

La reivindicació d'aquestes llengües de signes com a instruments de comunicació propis sorgeix en **l'últim quart del segle XX**. Es duen a terme nombroses trobades que inicien el debat sobre la necessitat del reconeixement i ús d'aquestes llengües per garantir el ple accés de les persones sordes i mutes a l'educació i la societat.

2.4 ESTRUCTURA

Com qualsevol idioma, la llengua de signes consisteix en paraules ordenades lògicament formant oracions, però a diferència de les llengües parlades, aquesta utilitza diferents signes estàtics o amb moviment per comunicar cada paraula.

(Per comoditat tots els exemples i els noms dels signes estan traduïts al català)

ASPECTES SEMÀNTICS

La paraula i el signe realitzat poden tenir una relació més o menys propera al concepte que representen. Depenent d'aquesta relació es poden classificar els signes en tres categories, de relació motivada, intermèdia o arbitrària.

Signes de relació motivada o icònica

El signe té relació amb l'objecte que representa, tant sigui en la forma del signe, el moviment o la relació espacial. Per exemple el signe CASA representa la forma de la teulada d'una casa. El signe COTXE representa la forma d'un volant d'un cotxe.

Signes de relació intermèdia

Els signes es basen en l'alfabet dactilològic (Representació de cada lletra de l'abecedari amb signes) per representar una paraula. Per exemple, els noms de les ciutats Barcelona i Mataró es formarien representant les lletres de l'abecedari que les componen, una per una. També hi han abreviacions locals enteses per un grup petit de gent, per exemple es pot acordar que Barcelona es representarà amb només la lletra B de l'alfabet dactilològic.

Signes de relació arbitrària

No tenen cap relació amb l'objecte que representen. No existeix cap motivació visual i el signe existeix perquè els usuaris de la LS ho han decidit així a través dels acords acadèmics en la Confederación Nacional de Sordos de España (CNSE). Per exemple, el signe BLAU no representa cap qualitat del color blau.

ASPECTES MORFOLÒGICS

La morfologia s'encarrega de l'anàlisi i els canvis de les paraules. Distingim entre:

Articles

No existeixen els articles, ja que no serien eficients a l'hora de signar.

Ex: "Quedem a les sis" = NOSALTRES-CITA-HORA-SIS.

Substantius

Els substantius es signen segons la seva forma (ex. AMPOLLA), la seva funció (ex. BATEDORA), la seva manera d'utilitzar-se (ex. s'utilitzen diferents signes per indicar la baixada d'una finestra del cotxe de manovella o automàtica), segons una part significativa (ex. LÀMPARA) o assenyalant-lo directament (ex. ULLS).

Verbs

Els verbs es signen fent referència primer al temps corresponent i després es signa el verb. S'utilitza el signe PASSAT per passat, AVUI per present, encara que si no s'indica el temps es suposa que estem parlant del present, i FUTUR per futur. Per comunicar unitats de temps s'utilitza el signe corresponent seguit d'un numeral.

Per exemple:

"Jo vaig menjar" = JO-PASSAT-MENJAR.

"He menjat a les vuit" = JO-HORA-VUIT-MENJAR.

Adjectius

Els adjetius es situen després del nom que accompanyen.

Ex: "La casa és blanca" = CASA-BLANCA.

Adverbis

Els adverbis es soLEN col·locar al final de l'oració complementant el significat del verb que accompanyen.

Ex: "La meva casa és a prop" = CASA-MEVA-A PROP.

Pronoms

Els pronoms funcionen igual que a la llengua parlada, es situen a l'inici de l'oració.

Ex: "Ell ve a les vuit" = ELL-HORA-VUIT-VENIR

Preposicions

S'utilitzen algunes preposicions (amb, contra, fins, per [hi ha distinció entre *por* i *para* en el LSE], sense), però el seu ús s'adapta a l'estructura de la llengua de signes, situant-les darrere de les paraules que accompanyen.

Ex: "Vull un cafè sense sucre" = JO-VOLER-CAFÉ-SUCRE-SENSE

Interjeccions

Les interjeccions es manifesten amb l'expressió de la cara i moviments gestuals.

Gènere

El gènere normalment només es manifesta quan es tracta d'un substantiu femení, suposant-se que és masculí quan no s'especifica. S'expressa amb el signe HOME o DONA darrere del substantiu.

Ex: "La meva germana és guapa" = MEU-GERMÀ-DONA-GUAPO

Nombr

Es pot expressar el plural de diferents formes: Repetint dues vegades el mateix signe, amb l'ús de les dues mans pel plural i una pel singular o afegint després del substantiu un signe quantificador definit (DOS, VUIT) o indefinit (MOLTS, ALGUNS).

ASPECTES SINTÀCTICS

En la LSE la relació entre signes s'organitza seguint una seqüència cronològica.

L'ordre dels signes segueix l'ordre dels esdeveniments, per exemple:

"Et vaig comprar un gelat després de menjar" =

JO-MENJAR-FÍ-DESPRÉS-JO-GELAT-COMPRAR

Hi han vàries formes de relacionar diferents conceptes entre ells.

Relació de causalitat

Quan dues oracions estan relacionades causalment s'utilitzen els signes PERQUÈ, MOTIU, PER i CULPA.

Per exemple, “Estic marejat perquè he begut” = JO-MAREJAR-MOTIU-BEURE

Relació de conseqüència

No hi ha cap signe específic per explicar les conseqüències d'una acció, però s'utilitza el signe PER AIXÒ.

Ex: “El control està suspès, repeteix-lo” = CONTROL-SUSPÈS-PER
AIXÒ-TU-REPETIR

Relació de contrast

Per expressar relació de contrast entre dues oracions s'utilitzen els signes PERÒ, IGUAL i QUE.

Ex: “M'agrada aquest cotxe però és molt car” =
AQUEST-COTXE-JO-AGRADAR-PERÒ-CAR

Relació condicional

La partícula condicional “si” de la llengua oral es representa amb el signe EXEMPLE.

Ex: “Si guanyo la loteria, em compraré un cotxe” =
EXEMPLE-JO-LOTERÍA-GUANYAR-COTXE-COMPRAR

Relació de finalitat

S'utilitzen els signes PER O PERQUÈ (en LSE, PARA o PARA QUE) per comunicar o preguntar la finalitat d'una acció.

Per exemple:

(En LSC) “Perquè has vingut?” = TU-VINDRE-PERQUÈ?

(EN LSE) “¿Para qué has venido?” = ¿TÚ-VENIR-PARA QUÉ?

2.5 ALFABET DACTILOLÒGIC

L'alfabet dactitològic és utilitzat normalment per comunicar noms propis, nombres, o paraules per les quals no hi ha signe. És necessari lletrejar cada lletra de la paraula seguint una taula amb els signes corresponents.

Existeixen dos tipus d'aquest:

- El sistema dactitològic visual o a l'aire:** Usat per les persones sordes, es realitza a l'aire i és captat visualment.
- El sistema dactitològic tàctil o al palmell:** Només usat per persones sordcègues, consisteix en lletrejar cada lletra al palmell de la mà del receptor, aquest pot captar els signes a través del tacte i reconstruir la paraula.

En aquest projecte, s'utilitza el sistema visual. Aquest consisteix en els següents signes:



2.6 ÚS EN EL PROJECTE

Com ja s'ha pogut observar, la llengua de signes no és tan fàcil com sembla en un principi. Per això aquest projecte no pot arribar a traduir tots els matisos i gramàtica d'aquesta llengua.

En el projecte s'intentarà detectar únicament els signes estàtics de l'alfabet dactilològic espanyol. Aquest inclou també els sons Ñ, RR, LL, i CH, donant un total de 30 sons, als que si descartem els signes amb moviment ens deixen amb un total de 18 signes. A la següent taula es poden veure totes les formes de les mans i les seves corresponents lletres que podrà detectar el programa:



Signes que es podran detectar

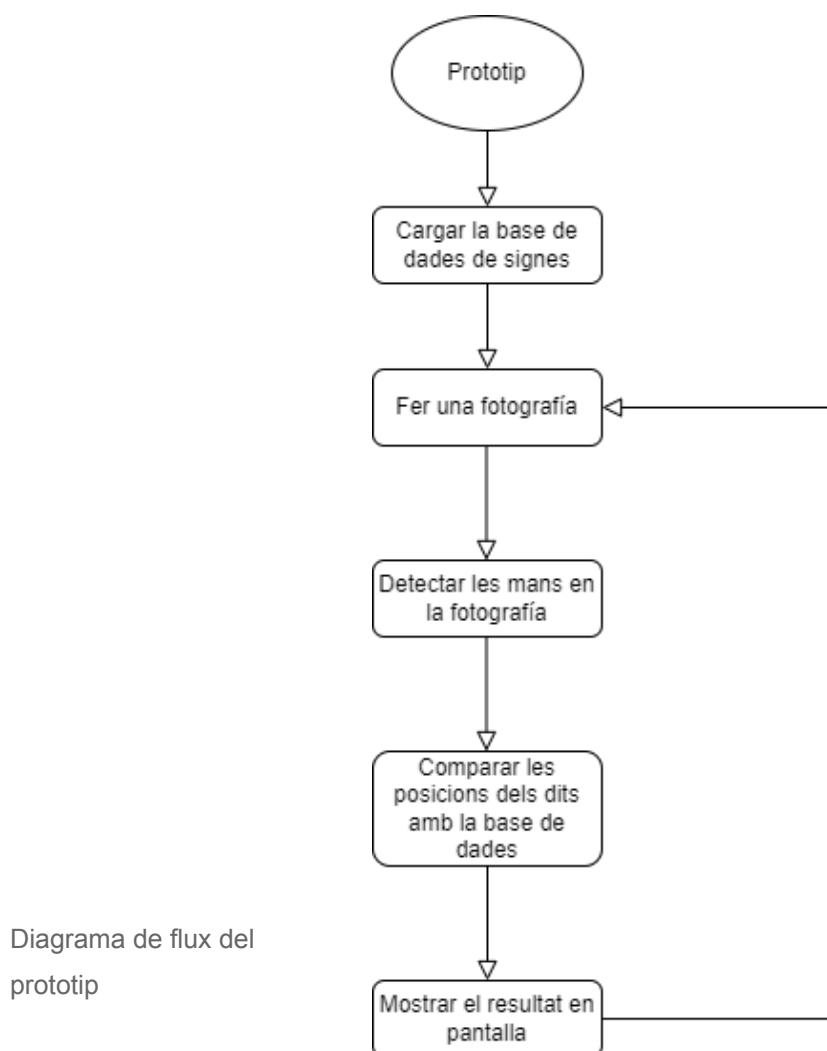
3. PLANIFICACIÓ

El pla inicial és crear un prototip que pugui detectar uns pocs signes i mostri els resultats en pantalla, després es passarà tot a una aplicació mòbil i s'afegiran els signes restants a la base de dades.

3.1 PLA DEL PROTOTIP

El prototip només serà funcional amb ordinador, consistirà d'una aplicació d'escriptori on es mostrarà la càmera en un costat, un gràfic de la detecció de mans a l'altre costat i el nom del signe que es creu que s'està realitzant.

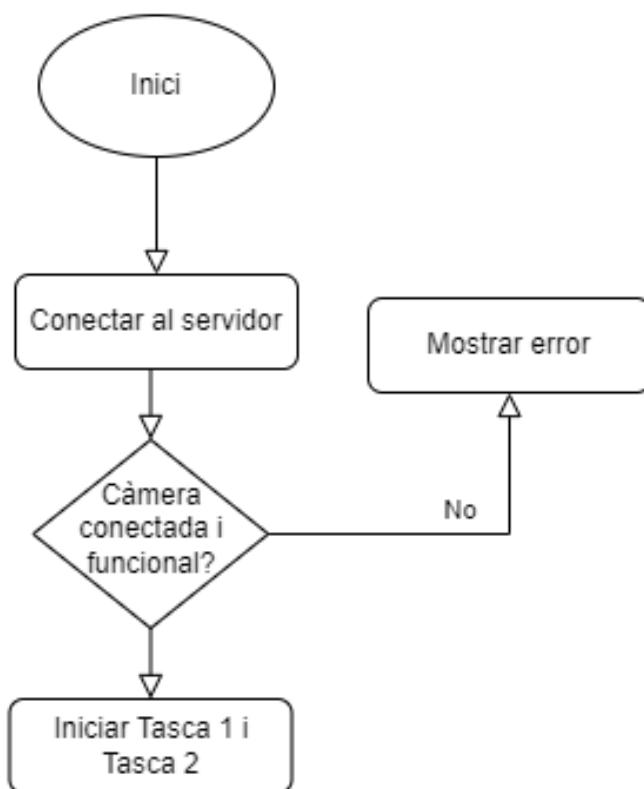
A continuació hi ha un diagrama de flux del funcionament del prototip:

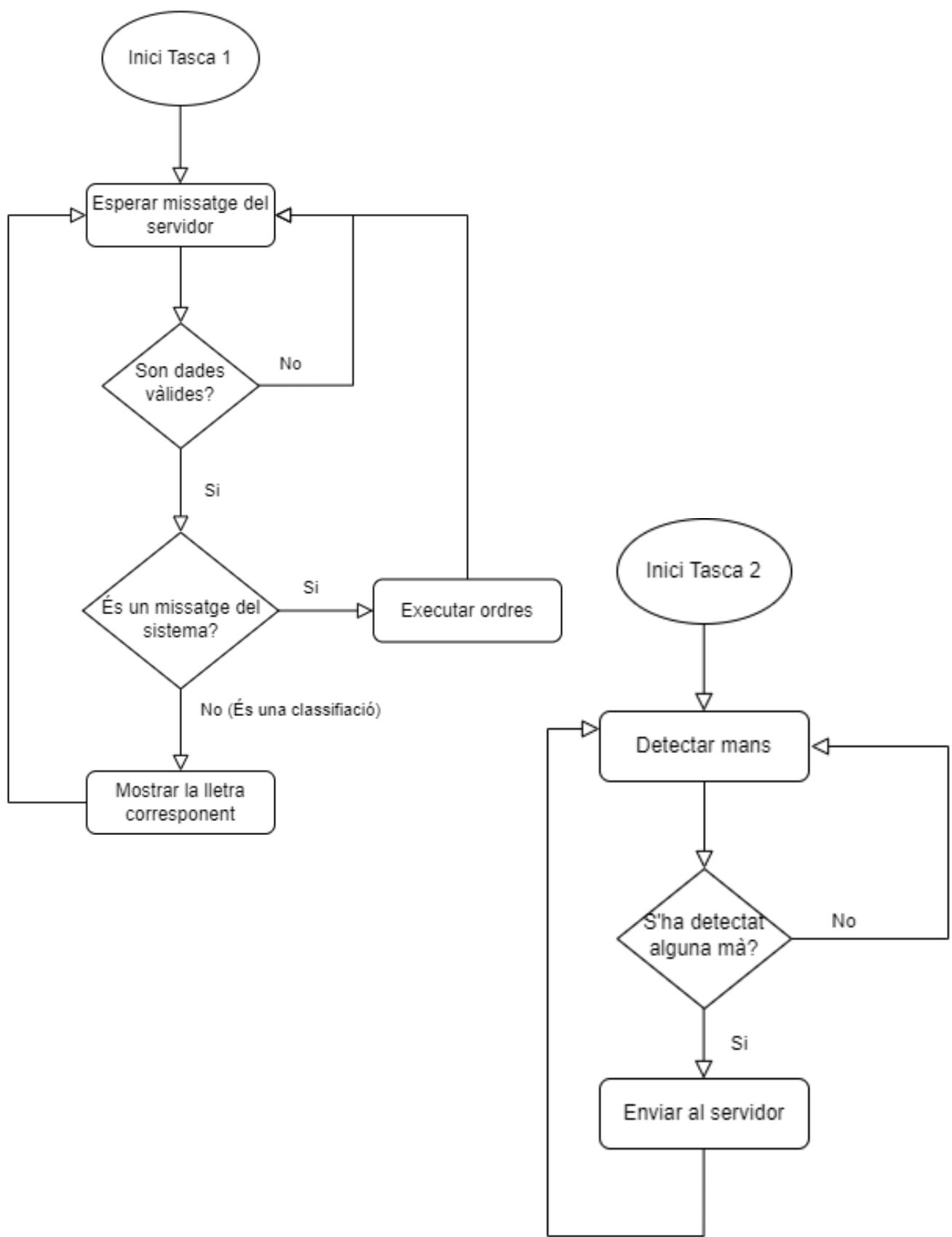


3.2 PLA DE L'APLICACIÓ FINAL

L'aplicació consistirà en una web que es comunicarà amb un servidor extern per realitzar la detecció de mans. Anomenarem al mòbil de l'usuari *client*. Al client capturarà cada fotograma de la càmera, es detectaran les posicions dels dits i s'enviaran al servidor. El servidor retornarà la classificació de la mà i el client la mostrerà en pantalla.

Dins del client es crearan dues tasques, a continuació es poden veure els diagrames de flux de l'aplicació.





Diagrammes de flux de l'aplicació final

4. LA INTEL·LIGÈNCIA ARTIFICIAL

La intel·ligència artificial és una branca de la computació que busca simular una conducta intel·ligent similar a la humana. Per aconseguir-ho s'utilitzen diferents algoritmes i mètodes, dels quals un és la xarxa neuronal artificial.

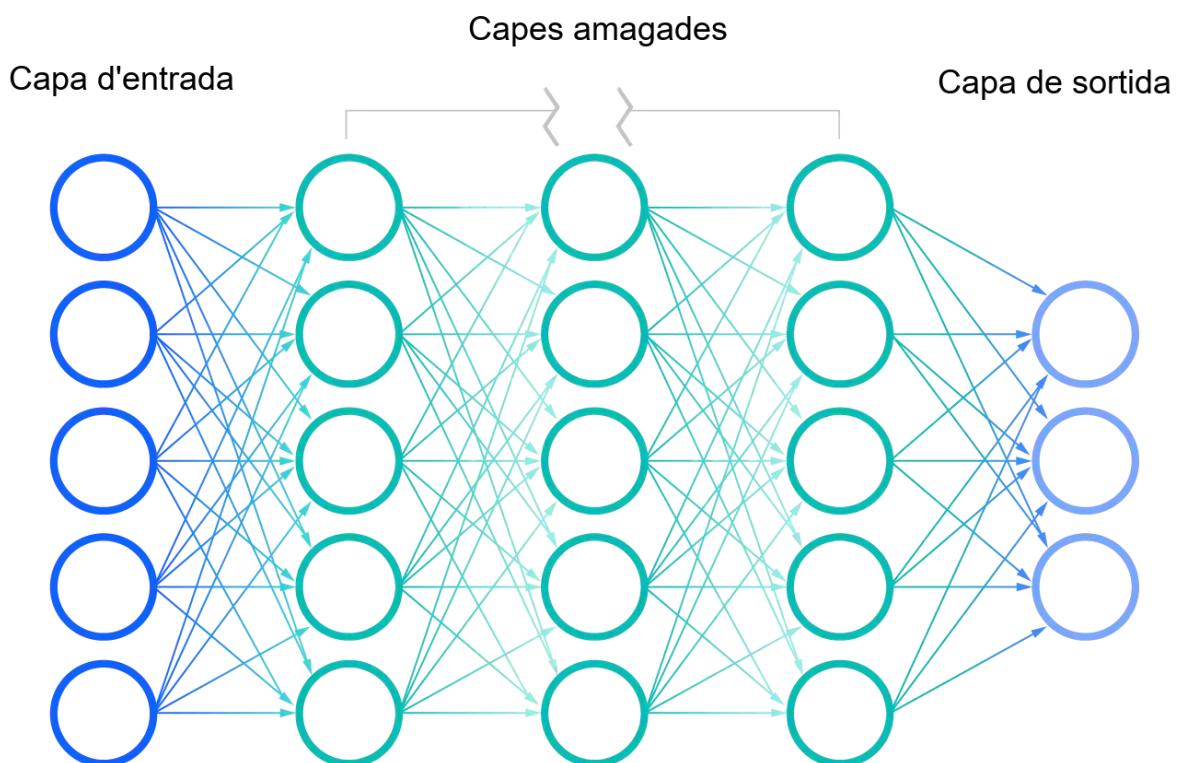
4.1 QUÈ ÉS UNA XARXA NEURONAL?

Una xarxa neuronal és un algoritme d'aprenentatge profund l'estructura del qual està inspirat en el cervell humà.

La seva estructura consisteix en capes de nodes, una capa d'entrada, vàries capes no visibles i una capa de sortida. Cada node, o neurona artificial, connecta amb un altre i té un pes i llindar associat. Si el valor de sortida d'algun node individual és més gran que el llindar especificat, aquell node és activat, enviant dades a la següent capa de la xarxa. Si no, cap dada s'envia a la següent capa.

Cada node de la xarxa té la seva pròpia esfera de coneixement, incloent-hi regles amb les que s'ha programat i regles que ha après per si mateix.

La primera capa rep unes dades d'entrada similars a com el nervi òptic rep informació sense processar. Cada capa rep la informació que ha processat la capa anterior, la processa i la passa a la següent, així successivament fins a arribar a la capa de sortida, que processa les dades finals.



Estructura general d'una xarxa neuronal

En entrenar una xarxa neuronal es canvien els valors de pes i llindar dels nodes, seguint un algoritme, perquè utilitzant unes dades d'entrada, s'aconsegueixin les dades de sortida que més s'acosten al resultat que es vol aconseguir. Això es pot fer amb un sistema de recompenses, fent servir dades d'entrenament o una simulació, es poden utilitzar diferents mètodes, entre els més populars hi ha:

- **Punts per progrés:** S'atorgaran punts a l'algoritme d'evolució dependent de l'avanç que arribi a fer la xarxa en un context que es pugui mesurar linealment. Per exemple, els metres que ha avançat un cotxe en una cursa.
- **Punts per semblança:** Aconseguirà punts en funció de la semblança del resultat de la xarxa neuronal i les dades d'entrenament. Per exemple, una xarxa neuronal que intenta endevinar la fondària d'un llac, **o la posició dels dits en una mà.**

- **Punts per objectius:** Uns definits objectius que ha d'aconseguir la xarxa atorgaran uns definits punts per cada objectiu. Per exemple, una xarxa que s'especialitza en endevinar el nom del color més prominent d'una imatge, en tenir només una limitada quantitat de noms, si aquesta l'endevina es marcarà l'objectiu com a complet. És preferible utilitzar un dels mètodes anteriors per obtenir una millor precisió d'entrenament.

4.2 TIPUS DE XARXES

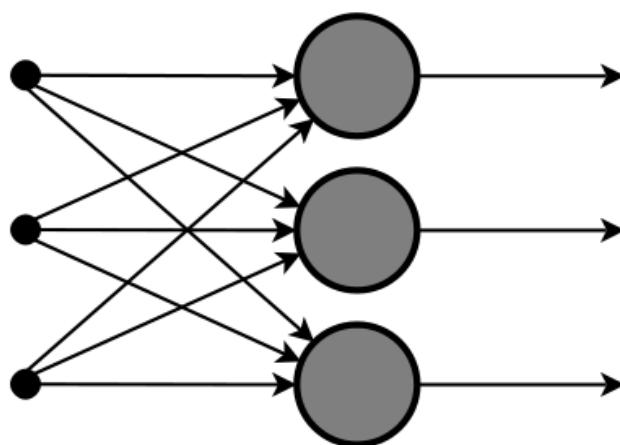
Hi ha molts tipus de xarxes neuronals, però només explicaré els més utilitzats:

XARXA DE RETROALIMENTACIÓ

Aquest és un dels tipus més senzills de xarxa neuronal. En una xarxa de retroalimentació, les dades passen per diferents nodes fins que arriben als nodes de sortida. Les dades es mouen en només una direcció des de la capa d'entrada fins a la de sortida. Això també és conegut com a una ona de propagació frontal, que normalment s'aconsegueix fent servir una funció d'activació classificatòria.

En comparació amb altres tipus de xarxa, aquest no té cap retropropagació i les dades només es mouen en una direcció. Aquest tipus de xarxa pot tenir una sola capa o diverses capes amagades.

En una xarxa de retropropagació, la suma dels productes dels nodes i els pesos són calculats, llavors són donats a la capa de sortida. Aquest és un exemple d'una xarxa de retroalimentació d'una sola capa:



Estructura d'una xarxa de retroalimentació

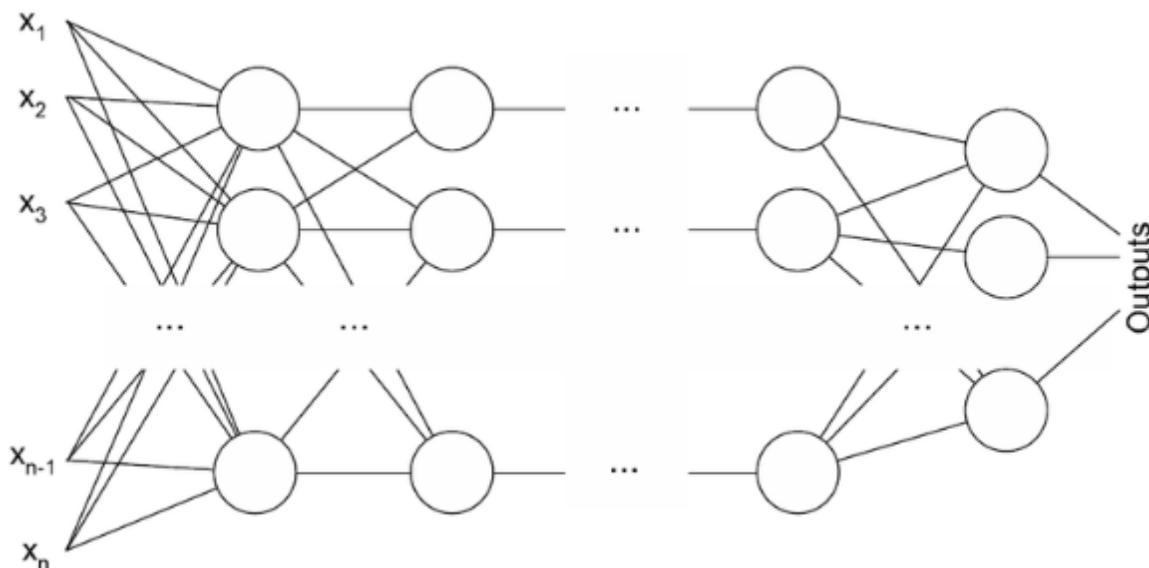
Aquest tipus de xarxa s'utilitza normalment en tecnologies de reconeixement facial, ja que la classificació segons les categories de sortida és complicada en aquestes aplicacions.

Una xarxa de retroalimentació simple està equipada per tractar amb dades que contenen molt de soroll (pertorbacions o turbulències en les dades).

PERCEPTRÓ MULTICAPA

Un perceptró multicapa té tres o més capes. S'utilitza per classificar dades que no es poden separar linealment. És un tipus de xarxa neuronal que està completament connectada, cada node de la capa està connectat a tots els nodes de la següent capa.

Així es veuria l'estructura d'un perceptró multicapa:



Estructura d'un perceptró multicapa

Aquest tipus de xarxa neuronal s'utilitza en tecnologies de reconeixement de veu i traducció automàtica.

XARXA NEURONAL CONVOLUCIONAL

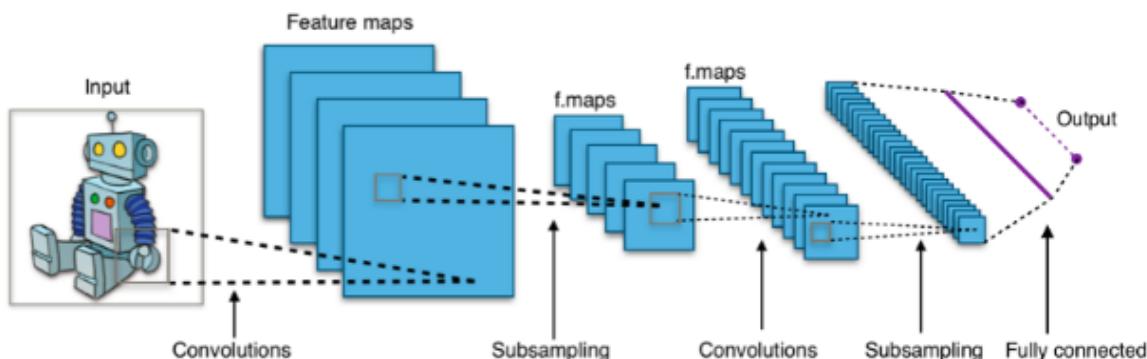
Una xarxa neuronal convolucional (CNN) fa servir una variació del perceptró multicapa. Aquesta conté una o més capes convolucionals, les quals poden estar completament interconnectades o agrupades.

Abans de passar les dades processades a la següent capa, la capa convolucional utilitza una operació convolucional en les dades. A causa a aquesta operació, la xarxa es pot estendre molt més amb molts menys paràmetres.

Gràcies a aquesta habilitat, aquestes xarxes tenen una gran eficàcia per al reconeixement d'imatges i vídeos, per exemple el model utilitzat en aquest projecte; processament de llenguatge natural, utilitzat en els traductors, i sistemes recomanadors, com el famós algoritme de YouTube.

També s'estan començant a utilitzar en l'anàlisi d'imatges en el camp de l'agricultura, on dades sobre el temps són extretes dels satèl·lits.

Aquesta és una imatge il·lustrant l'estructura d'una CNN:

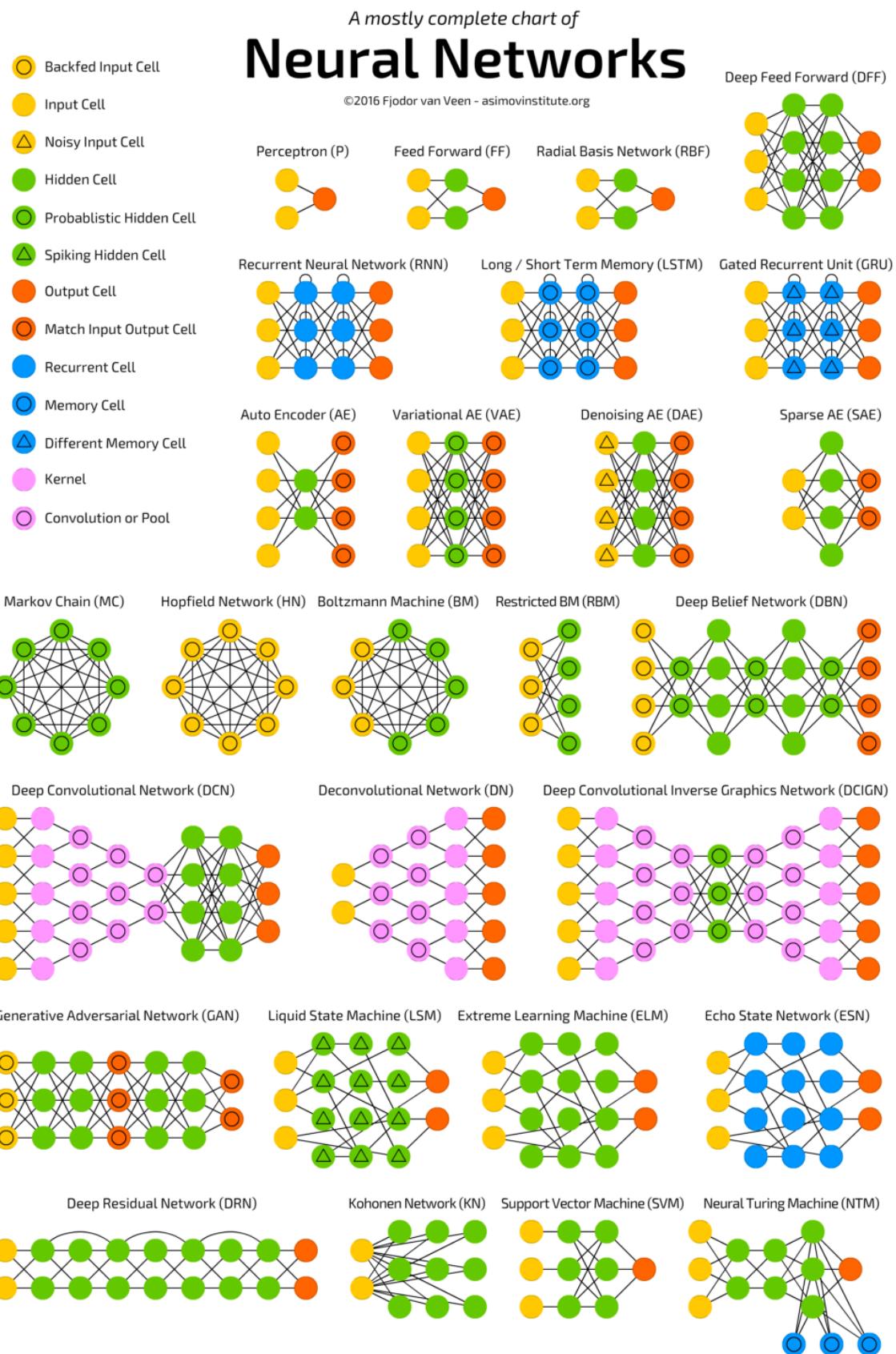


Estructura d'una CNN

ALTRES

Aquest tres són els tipus més usats i coneguts. Tanmateix, també hi ha d'altres als que val la pena donar una ullada encara que no siguin tan populars.

Aquest esquema, encara que està en anglès, mostra tots els tipus importants de xarxa neuronal.



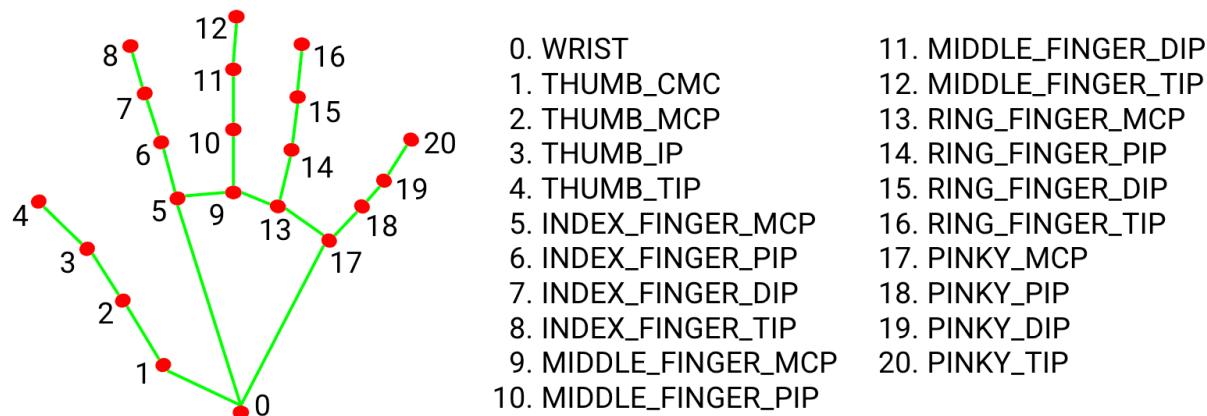
Esquema de tots els tipus de xarxes neuronals

5. DETECCIÓ

Com ja s'ha dit prèviament, la detecció consta de dues parts. La detecció de la posició dels dits i la classificació del signe.

5.1 POSICIÓ DELS DITS

Per detectar les posicions dels dits s'ha fet servir un model de xarxa neuronal pre entrenat anomenat [Media Pipe Handpose](#), basat en [Tensorflow](#). A aquest li pots donar una imatge d'entrada i et retornarà varis valors entre els quals hi ha la posició dels dits, seguint l'ordre de la imatge de sota:



Esquema de les posicions dels dits retornades

El model en el llenguatge JavaScript (Utilitzat pel programa final) retorna quatre valors:

1. Puntuació: Varia de 0 a 1. És la confiança del model en la mà detectada.
2. Mà: Pot ser “right” o “left”. És la prediccio de quina mà s'està utilitzant.
3. Punts: Llista etiquetada de totes les posicions dels trets més rellevants de la mà en relació amb les coordenades dels píxels de la pantalla.
4. Punts 3D: Llista etiquetada de posicions en un espai 3D en metres.

Aquest és un exemple dels resultats que podria donar una detecció:

```
[  
  {  
    score: 0.8,  
    handedness: 'Right',  
    keypoints: [  
      {x: 105, y: 107, name: "wrist"},  
      {x: 108, y: 160, name: "pinky_finger_tip"},  
      ...  
    ],  
    keypoints3D: [  
      {x: 0.00388, y: -0.0205, z: 0.0217, name: "wrist"},  
      {x: -0.025138, y: -0.0255, z: -0.0051, name: "pinky_finger_tip"},  
      ...  
    ]  
  }  
]
```

Possibles dades retornades

I aquesta imatge il·lustra perfectament la conversió que fa el model a dades en 3D:



Exemple de la detecció 3D de la mà

Pel prototip he utilitzat una [versió escrita en Python](#) que no retorna dades com la puntuació o quina ma creu que s'està utilitzant. Retorna dues llistes de punts, en espai de pantalla i en espai real, sense etiquetar.

5.2 CLASSIFICACIÓ

Per a la classificació no he utilitzat intel·ligència artificial, sinó un algoritme classificador anomenat “Classificador Bayesià Ingenu Gaussià”, aquest funciona utilitzant el Teorema de Bayes.

5.2.1 EL TEOREMA DE BAYES

El Teorema de Bayes diu que:

$$P(A | B) = \frac{P(B | A) \cdot P(A)}{P(B)}$$

I que

$$P(B) = P(B | A) \cdot P(A) + P(B | \text{no } A) \cdot P(\text{no } A)$$

Per entendre aquesta equació necessitem saber que significa cada símbol, tots aquests són probabilitats, de les quals hi ha varis tipus:

- **Probabilitat marginal:** Probabilitat d'un esdeveniment sense tenir en compte altres variables aleatòries. Ex: $P(A)$.
- **Probabilitat conjunta:** La probabilitat de dos o més esdeveniments simultanis. Ex: $P(A \text{ i } B)$ o $P(A, B)$.
- **Probabilitat condicional:** Probabilitat d'un o més esdeveniments donat que succeeix un altre. Ex: $P(A \text{ donat } B)$ o $P(A | B)$.

La probabilitat conjunta pot ser calculada utilitzant la probabilitat condicional:

$$P(A, B) = P(A | B) \cdot P(B)$$

Aquesta també compleix amb la propietat commutativa:

$$P(A, B) = P(B, A)$$

La probabilitat condicional es pot calcular utilitzant el **Teorema de Bayes** o la probabilitat conjunta:

$$P(A | B) = \frac{P(A, B)}{P(B)}$$

Però no és simètrica:

$$P(A | B) \neq P(B | A)$$

Un bon exemple perquè s'entengui el seu funcionament és l'anàlisi d'un test mèdic. Considerarem un grup d'humans que pot tenir o no càncer i un test per detectar-lo que pot sortir positiu o negatiu. Quina és l'eficàcia del test? És a dir, quan el test és positiu, quina probabilitat hi ha que el pacient realment tingui càncer.

Utilitzarem la sigla T per dir que el test és positiu i -T per indicar quan és negatiu, el mateix pel càncer.

Tenim les següents dades:

1. La sensibilitat del test, que és 85%. Això significa que quan el pacient té càncer, el test dona positiu el 85% de les vegades:

$$P(T | C) = 85\%$$

2. L'especificitat del test, 95%. Quan una persona **no** té càncer, el test donarà negatiu el 95% de les vegades.

$$P(-T | -C) = 95\%$$

3. La probabilitat que una persona tingui càncer, que és bastant baixa, així que li assignarem 0.02%.

$$P(C) = 0.02\%$$

Podem substituir una de les fórmules per:

$$P(C | T) = \frac{P(T | C) \cdot P(C)}{P(T)}$$

Però no tenim el valor de $P(T)$.

$P(T)$ és la probabilitat que en general el test sigui positiu, la podem calcular utilitzant la segona fórmula:

$$P(B) = P(B | A) \cdot P(A) + P(B | \text{no } A) \cdot P(\text{no } A)$$

Que substituïda és:

$$P(T) = P(T | C) \cdot P(C) + P(T | \neg C) \cdot P(\neg C)$$

Les probabilitats $P(T | \neg C)$ i $P(\neg C)$ es poden calcular fent el complement dels seus contraris:

$$\begin{aligned} P(\neg C) &= 100 - P(C) = 100 - 0.02 = 99.98\% \\ P(T | \neg C) &= 100 - P(\neg T | \neg C) = 100 - 95 = 5\% \end{aligned}$$

Llavors ara podem substituir per trobar el valor de $P(T)$.

$$P(T) = 85\% \cdot 0.02\% + 5\% \cdot 99.98\% = 5.016\%$$

I tornem a substituir a la primera equació:

$$P(C | T) = \frac{85\% \cdot 0.02\%}{5.016\%} \simeq 0.34\%$$

En conclusió, la prova és força menys eficaç del que pot semblar sense els càlculs.

Ara que s'ha entès el teorema de Bayes i el concepte de probabilitats, es pot passar a explicar el classificador de Bayes.

5.2.2 CLASSIFICADOR DE BAYES INGENU

En el prototip i l'aplicació he utilitzat la mateixa llibreria de classificació que he fet en Python. Aquesta implementa el classificador de Bayes ingenu amb el tipus de dades que m'ha sigut més adient per la base de dades.

Per poder classificar les mans es necessita una base de dades amb tots els tipus de símbols prèviament gravats.

L'estructura de la base de dades consta d'una carpeta amb arxius .hand etiquetats dependent del signe, tal com mostra la imatge següent:

📄 r_let_a.hand	17/10/2022 18:44	Fitxer HAND	109 kB
📄 r_let_b.hand	19/10/2022 18:42	Fitxer HAND	109 kB
📄 r_let_c.hand	19/10/2022 18:42	Fitxer HAND	110 kB
📄 r_let_l.hand	17/10/2022 18:57	Fitxer HAND	109 kB
📄 r_let_m.hand	19/10/2022 18:42	Fitxer HAND	109 kB
📄 r_let_o.hand	17/10/2022 18:59	Fitxer HAND	109 kB
📄 r_let_s.hand	19/10/2022 18:42	Fitxer HAND	109 kB
📄 r_let_u.hand	17/10/2022 18:39	Fitxer HAND	109 kB

Carpeta de la base de dades

Cadascun d'aquests arxius conté 100 captures de les posicions de dits fetes mentre es realitza el signe corresponent i es mou la mà lentament per afegir irregularitats.

Cada captura està separada per una nova línia i cada valor de posició està separat per un espai.

```

1 1.1 1.1 0.29683522513338223 0.637389167941803 0.9651745270450306 0.49894560384162256 0.3465259790286626 0.8719158006927791 0.7847269440577233 0.161
2 1.1 0.8507519086912932 0.3640406914215588 0.5323071925156152 0.9013418678183079 0.6514348418581202 0.307310934303457 0.9951183774196036 0.88541746
3 1.1 1.1 0.36107209368514015 0.531437403797435 1.0346119191659158 0.6501267894955082 0.33649066004304173 1.051887650689921 0.8806362321216028 0.204
4 1.1 1.0754928076718167 0.3915908300121772 0.5821866553686607 0.103344043686734151 0.6766682178218594 0.3506118700395809 1.0025039957378796 0.89223284
5 1.1 1.0347454947834336 0.4023474230913685 0.571090884226286 0.9957951321045527 0.669514765894639 0.3274232251544061 0.9603857097692065 0.89038741
6 1.1 1.0030921262352566 0.4341106248144553 0.5473773384274094 0.9816703857188397 0.7140086608636501 0.3294024828751423 0.9561683136754862 0.90801412
7 1.1 0.9307721022833323 0.394934906287522 0.6257092508902261 0.9282834567286754 0.6788524020313879 0.3804321518266454 0.9332607478379892 0.8941489
8 1.1 0.9403766921718658 0.42269985643314834 0.5916532262255563 0.9198177575044144 0.7191812376682145 0.3727098016286581 0.911183048852266 0.91104432
9 1.1 0.9281904537140834 0.4870278231213563 0.535246881099407 0.9187262965366688 0.7058801951077073 0.2900876853538645 0.9187503687533187 0.921665581
10 1.1 1.0080514815923376 0.4174531833736348 0.5460366017211118 0.9481891649268624 0.7872038167047411 0.3489477178124278 0.93573778030529782 0.91765295
11 1.1 0.98735088353221957 0.4198170433679099 0.5508686024113427 0.937788330548926 0.7004961047707228 0.3366432518580974 0.920525050506658711 0.90243018
12 1.1 0.9730684980166953 0.4276127458466463 0.500812776927227 0.9223906222248535 0.7131997886282657 0.2919334785253276 0.920024971878515 0.90976818
13 1.1 1.047583137594264 0.39672509003334332 0.5388971484280306 1.010001236246755 0.6757109718297221 0.3383725686692909 0.9724193348992458 0.8943209589
14 1.1 1.0615108250804461 0.3702308506244695 0.5481397237986115 0.988031491143116 0.6495734297937427 0.29925920453030636 0.965537817488813 0.88819658
15 1.1 1.061557121880432 0.4051713751634893 0.5574239201386262 1.007221187287756 0.6913980121930491 0.5754405074645992 0.89993313
16 1.1 0.985722782864399 0.3924078678497197 0.5697531944617303 0.10534852636503376 0.6895962977009985 0.35720233802201773 1.0248793958430796 0.90741762
17 1.1 1.0499909839514336 0.3831749972877331 0.5825941119953713 0.9884414257378132 0.6678449757202723 0.3652411379643029 0.9793805163190479 0.883560955
18 1.1 1.1 0.4094298366181727 0.202225375206398 0.6965801167780512 0.3732226524333173 0.8903755918046216 0.248
19 1.0701277671748803 0.8800429630656228 0.34103552258443326 0.5653672397757616 0.9392349802580488 0.5797907538673346 0.324619498419878 0.928147737598
20 1.099734393756031 0.9720614105229489 0.3831110317538493 0.5859409978034886 0.8560637560637561 0.6518235355873901 0.40661918967739075 0.837299429607
21 0.9379225473136512 0.456555376166471 0.403594462765083 0.41434525615015964 0.5023424174657923 0.6785223896935254 0.2433987622768802 0.63151650684
22 0.9448753429290628 0.46496204916399475 0.36874749884988367 0.4470217865227366 0.48293208702425594 0.6272612811498451 0.25660036356971216 0.57510899
23 0.8057982387370134 0.5345989179051422 0.32787349166958263 0.38380412058537205 0.4978180861138985 0.581607104755466 0.20746536920891784 0.56624175
24 0.7985974123127609 0.34532757698197447 0.33519526628149365 0.36643808039833243 0.3860648327521633 0.5928963167008005 0.18422457168604223 0.483922805
25 0.8339185861208626 0.49282163624070316 0.33065889785296654 0.3894404718845659 0.47146720757268423 0.57555656002042583 0.2286275654265665 0.531778228
26 0.7924226669331372 0.20140204487259836 0.32391776805073 0.3978841879930166 0.27231006141780223 0.5606434121015682 0.20875618872992882 0.3886278582
27 0.8245336269299651 0.1 0.3169450681318184 0.41966289248048827 0.274746658398787377 0.5587367972196331 0.2135596069301398 0.4723301639796599 0.88646
28 0.7669586023629461 0.1 0.29971380198359243 0.36505898650157191 0.24396405782002867 0.52488496516232 0.18221417415807884 0.4328558406642453 0.7779118
29 0.7767976256919499 0.1 0.2867115933825089 0.3994369320115846 0.25637380529813372 0.1961520835036863 0.45718655325099045 0.74861049
30 0.7999892670959435 0.1 0.2873858427650373 0.41535012312002395 0.2895479054334301 0.5000881338791284 0.21335604986342419 0.5034114475321444 0.736296
31 0.7308976860651977 0.1 0.30436501880725375 0.348765187361216 0.3006147540983607 0.5286311882947835 0.14697037182297523 0.5079918032786885 0.7791498
32 0.712672202441087 0.1 0.28573975881337843 0.3324284823414614 0.3187747035573123 0.5217492161666064 0.15691415524055724 0.51987111462450593 0.7768743
33 0.71849180691805475 0.1 0.28178918485062277 0.3466928578743684 0.30985540334855405 0.5115546371303908 0.15234467731402876 0.5279870624048706 0.77420
34 0.7403214255763577 0.1 0.28269065971579554 0.35406191278106275 0.26872037914691943 0.5138394694572884 0.16880524571676414 0.4785693127962085 0.79771
35 0.7812172778841047 0.1 0.25459239704748116 0.37283779221525437 0.2934571890145396 0.49129493836923444 0.1675180883756905 0.520889549273021 0.762758
36 0.7843341027798594 0.1 0.26756694631674455 0.3943878207875581 0.2823229986117538 0.4934472263314693 0.176546624686849583 0.5131189264229523 0.758155
37 0.82343900628691 0.1 0.25841869381239967 0.42958479184161296 0.2991960696739616 0.480214428172723 0.20676558402453077 0.5413242518981688 0.7442102
38 0.824051376212649 0.1 0.27861499279856494 0.4266491657538878 0.2428246013667426 0.5029162270444438 0.2953338947887171 0.45429954441913445 0.7605162
39 0.8110069995867373 0.1 0.3083437457566345 0.4273949386388427 0.277407315255036 0.5432321000238262 0.16722947495194979 0.4967404827071411 0.79785
40 0.7397767958903176 0.1 0.2935141471964512 0.38928781510281874 0.2145622492915762 0.5318114803410317 0.18037616678158946 0.34925477496043866 0.80668
41 0.7561715216656946 0.6715382646531866 0.2899396118237424 0.3933897035108993 0.5860506782697722 0.5519622752253036 0.20265303188933587 0.57606859482
42 0.78282015347150887 0.8176953676376235 0.31227060066795864 0.3838123948781883 0.7725266264596432 0.56315951609210532 0.1872998430103875 0.7227383549
43 0.78180426651338915 0.968039947170179 0.334524162507012 0.3635022093364204 0.8786797415332662 0.598152559950026 0.18275199136505327 0.81010163327
44 0.7504921008911214 1.1 0.3497139588502107 0.3666941070912234 0.882956362805745 0.599618288623896 0.17948309876271312 0.784715558601782 0.90812647
45 0.7514009340631032 1.1 0.33016599752343523 0.3635515552033195 0.9273153641860202 0.5796140580764371 0.18753173281601385 0.8020964005198621 0.873315
46 0.7717721518577826 1.1 0.34974770067401534 0.37177545795478295 0.9065089066499503 0.6082395416760212 0.2020254317374462 0.7539727315467795 0.911535

```

Continguts d'un arxiu de la base de dades

En el programa les noves línies es poden diferenciar fàcilment, ja que en programació es llegeixen com el valor “\n” (newline).

Anomenarem als noms de cada signe “classe”.

Ara podem aplicar el teorema de Bayes per obtenir la possibilitat que unes dades siguin d'una classe, substituint:

$$P(\text{classe}|\text{dades}) = \frac{P(\text{dades}|\text{classe}) \cdot P(\text{classe})}{P(\text{dades})}$$

Però més endavant la simplificarem. Però primer, s'explicarà el programa.

SEPARAR PER CLASSES

És necessari calcular la probabilitat de les dades segons les classes. Això significa que primer s'han de separar totes les dades per classes.

Aquestes tres funcions carreguen la base de dades i la separen per classes.

```
def loadDataset(dir): # Carga la base de dades

    dataset = {}

    for (dirpath, dirnames, filenames) in os.walk(dir):
        for i, filename in enumerate(filenames):
            if (filename.endswith(".hand")):
                f = open(os.path.join(dirpath, filename), "r")
                data = f.read().split("\n")
                f.close()

                dataset[i] = {"data": [], "label": ""}

                for dataPoint in data:

                    if (dataPoint == ""):
                        continue

                    dataset[i]["data"].append([float(z) for z in dataPoint.split(" ")]) # Convertir a float

                dataset[i]["label"] = filename.replace(".hand", "")

    return dataset
```

```
def deScrapeSet(dataset): # Re-estructura les variables, optimització
    newDataSet = []

    for dataClass in dataset:
        for dataPoint in dataset[dataClass]["data"]:
            newReg = dataPoint
            newReg.append(dataClass)
            newDataSet.append(newReg)

    return newDataSet
```

```
def separateByClass(dataset): # Separa dades per classe

    classified = {}

    for item in dataset:
        if (item[len(item) - 1] not in classified):
            classified[item[len(item) - 1]] = []

        classified[item[len(item) - 1]].append(item)

    return classified
```

Tot això ens deixa amb una base de dades separades en columnes amb la següent estructura:

[X1, X2, X3, X4... Xn, Classe]

On Xn són els valors de cada captura de la mà.

RESUM DE LA BASE DE DADES

Es necessiten dues estadístiques de la base de dades; la mitjana aritmètica i la desviació estàndard. Després veurem com aquestes dues variables són utilitzades per calcular la probabilitat.

La mitjana aritmètica és simplement la suma de tots els nombres introduïts entre el nombre de nombres:

```
def mean(numbers): # Mean (mitjana aritmètica)
    return sum(numbers) / float(len(numbers))
```

La desviació estàndard és la mesura de la dispersió de les dades en relació amb la mitjana. Es calcula com la mitjana de la diferència de tots els nombres amb la mitjana anterior.

```
def stdev(numbers): # Standard deviation (desviació estàndar)
    avg = mean(numbers)
    variance = sum([pow(x - avg, 2) for x in numbers]) / float(len(numbers) - 1)
    return math.sqrt(variance)
```

Es crea un resum de cada columna de la base de dades, o sigui, un resum per cada tipus de posició capturat (WRIST, THUMB_CMC, THUMB_MCP, THUMB_IP...). La següent funció retorna una llista de tots els resums, cadascun conté la mitjana aritmètica, desviació estàndard i nombre d'elements en la columna.

```
def summarizeDataset(dataset): # Retorna [(mean*(V1, V2, V3...), stdev*(V1, V2, V3...), len(V1, V2, V3...))]
    summaries = [(mean(attribute), stdev(attribute), len(attribute)) for attribute in zip(*dataset)]
    del summaries[-1]
    return summaries
```

Un exemple dels valors que podria retornar són (en el cas que només hi hagués dues columnes):

```
[(5.17833386499999, 2.7665845055177263, 10), (2.9984683241, 1.218556343617447, 10)]
```

Que visualitzat és:

5.17833386499999	2.9984683241
2.7665845055177263	1.218556343617447
10	10

I amb paraules seria:

Mitjana 1ra columna	Mitjana 2ra C
Desviació 1ra columna	Desviació 2ra C
Nombre d'elements 1ra columna	Nombre d'elements 2ra C

RESUM DE LES DADES PER CLASSE

Necessitem estadístiques de la base de dades ordenades per classe. Prèviament, ja hem creat dues funcions, una per separar les dades per classe i una altre per resumir la base de dades per cada columna.

Podem utilitzar les dues per separar els resums per classe:

```
def summarizeByClass(dataset):
    separated = separateByClass(dataset)
    summaries = {}

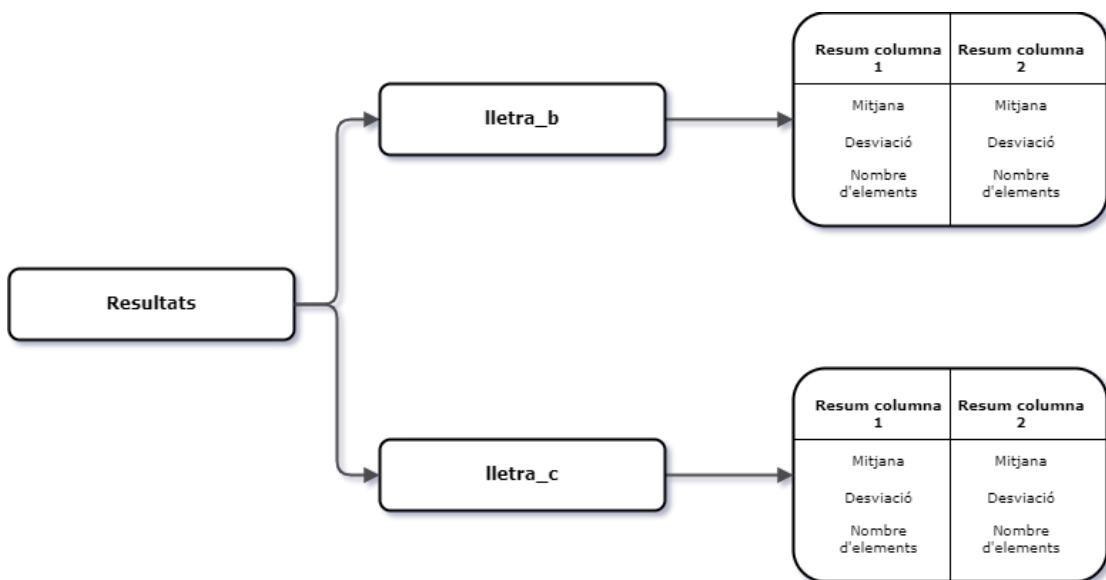
    for classValue, rows in separated.items():
        summaries[classValue] = summarizeDataset(rows)

    return summaries
```

Utilitzant aquesta funció tenim tots els valors separats en un diccionari de Python amb el que podem accedir al resum de cada classe. Per exemple:

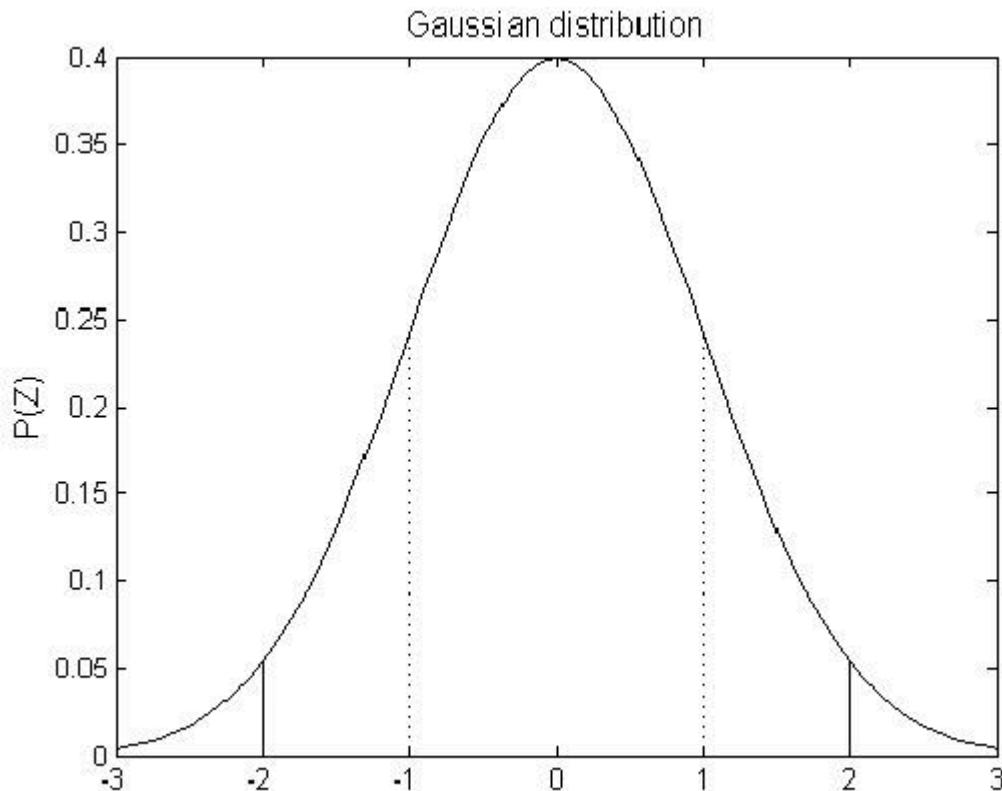
```
{“lletra_b”: [(5.17833386499999, 2.7665845055177263, 10), (2.9984683241, 1.218556343617447, 10)], “lletra_c”: [(3.7665845055177263, 4.17833386499999, 10), (6.218556343617447, 1.9984683241, 10)]}
```

Cadascun dels elements del diccionari conté el resums de cada columna de totes les dades de mostra etiquetades amb la lletra corresponent:



FUNCIÓ DE DENSITAT DE PROBABILITAT GAUSSIANA

Calcular la probabilitat que s'observi un valor X_n és difícil. Ho podem fer assumint que X_n surt d'una distribució de corba de campana o Gaussiana.



Una distribució Gaussiana es pot representar amb només dos nombres: la mitjana aritmètica i la desviació estàndard. Amb això podem estimar la probabilitat d'un valor donat. Aquesta fórmula es diu funció de densitat de probabilitat Gaussiana (o FDP Gaussiana), i es pot calcular de la següent manera:

$$f(x) = \frac{\sqrt{2\sigma^2} e^{-\frac{(-\text{mitjana}+x)^2}{2\sigma^2}}}{2\sqrt{\pi}}$$

On sigma és la desviació estàndard i “mitjana” la mitjana aritmètica.

Aquest és el codi de Python que la calcula:

```
def calculateProbability(x, mean, stdev): # Gaussian Probability Distribution Function / Gaussian PDF

    # f(x) = (1 / sqrt(2 * PI) * sigma) * exp(-((x-mean)^2 / (2 * sigma^2)))
    # sigma = Desviació estàndar d'X
    # mean = Mitjana aritmètica d'X

    exponent = math.exp(-(math.pow(x - mean, 2) / (2 * math.pow(stdev, 2))))
    return (1 / (math.sqrt(2 * math.pi) * stdev)) * exponent
```

PROBABILITAT DE CLASSES

Ara ja hem arribat al final del classificador. S'ha de calcular la probabilitat que unes dades estiguin en una classe, això es repetirà amb totes les classes i la probabilitat més gran serà la classe predicta.

Per utilitzar menys recursos i no realitzar operacions innecessàries, traurem la divisió del teorema de Bayes. Aquesta només normalitza els valors de 0 a 1 perquè tinguin definició de probabilitat. Sense aquesta, no estarem calculant la probabilitat de les dades en la classe, però el valor més gran continuarà sent la classe que més s'assembla a les dades introduïdes. La nova equació és:

$$P(\text{classe}|X_1, X_2 \dots X_n) = P(X_1|\text{classe}) \cdot P(X_2|\text{classe}) \cdot P(X_n|\text{classe}) \cdot P(\text{classe})$$

Aquesta funció retorna les probabilitats que una sèrie de dades estiguin en cadascuna de les classes possibles:

```
def calcClassProbabilities(summaries, row):
    total_rows = sum([summaries[label][0][2] for label in summaries])
    probabilities = {}
    for class_value, class_summaries in summaries.items():
        probabilities[class_value] = summaries[class_value][0][2] / float(total_rows)
        for i in range(len(class_summaries)):
            mean, stdev, count = class_summaries[i]
            probabilities[class_value] *= calculateProbability(row[i], mean, stdev)
    return probabilities
```

Ara només fa falta una funció que al donar-li una captura d'una mà, la base de dades i els resums retorni la classe en la qual és més probable que estigui aquell signe:

```
def predict(data, dataset, summary):
    probabilities = calcClassProbabilities(summary, data)

    bestProb = [-1, -1]
    for key, value in probabilities.items():
        if (value > bestProb[1]):
            bestProb[0] = key
            bestProb[1] = value

    return dataset[bestProb[0]]["label"], bestProb[1]
```

I una funció que inicialitzi el programa:

```
def load(datasetDir, debug=False):

    debugPrint("Loading dataset...", debug)

    dataset = loadDataset(datasetDir)
    summary = summarizeByClass(deScrapeSet(dataset))

    debugPrint("Loaded " + str(len(dataset)) + " one hand static signs", debug)
    debugPrint("Dataset uses " + str(sys.getsizeof(dataset)) + sys.getsizeof(summary)) + " bytes of ram", debug)
    debugPrint("Raw dataset: " + str(sys.getsizeof(dataset)) + " bytes", debug)
    debugPrint("Summary: " + str(sys.getsizeof(summary)) + " bytes", debug)

    return dataset, summary
```

La qual també imprimeix en pantalla informació sobre els signes carregats i els recursos que utilitzen.

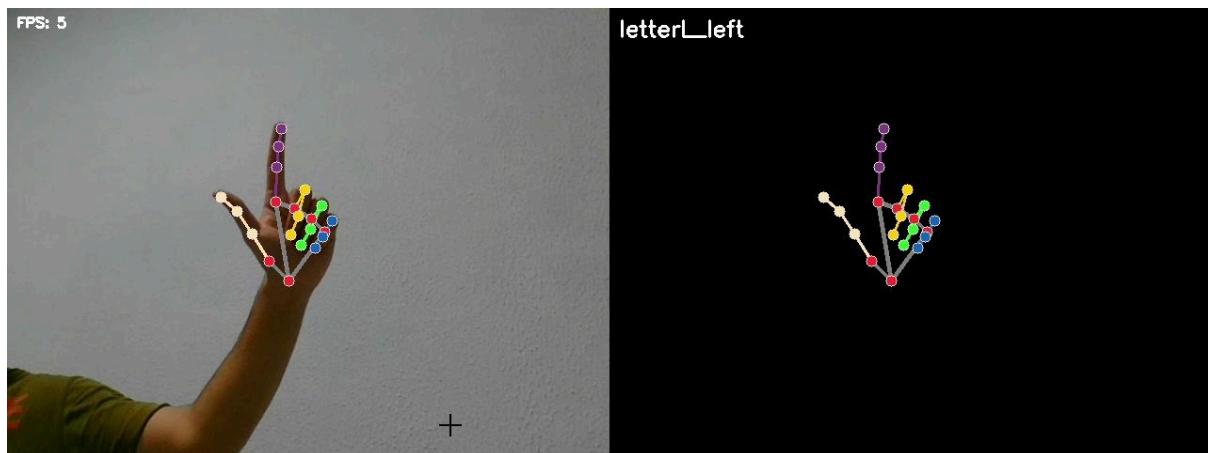
5.3 LIMITACIONS

Aquest mètode de classificació té vàries limitacions. La més important és la impossibilitat de detectar signes en moviment, ja que només pot classificar una captura dels dits, no una seqüència de captures.

Tampoc és capaç de classificar més d'una mà a la vegada. Es poden detectar vàries, amb molt poc rendiment, però no té sentit fer-ho.

6. PROTOTIP

Abans de fer l'aplicació final, feta en varis llenguatges de programació, es va fer un prototip només amb Python. Encara que el seu rendiment era bastant baix i tenia molts d'errors, això va permetre entendre l'estructura que tindria el programa final.



Versió final del prototip

7. APPLICACIÓ

El primer pas per la producció de l'aplicació és crear la part visible per l'usuari, després es pot programar el funcionament.

7.1 FRONTEND

Malgrat això, abans de començar a programar irreflexivament s'ha de dissenyar el frontend (part visible per l'usuari). A la següent imatge hi ha un esquema, subjecte a canvis, de la futura aplicació.



Disseny de l'aplicació

Ara que ja hi ha un pla, es pot començar a programar.

L'aplicació no és realment una aplicació d'Android nativa (escrita i exportada en el llenguatge de programació que utilitza el sistema operatiu), sinó que és una web també accessible per a l'ordinador; fent que el procés de prova i modificació sigui molt més fàcil.

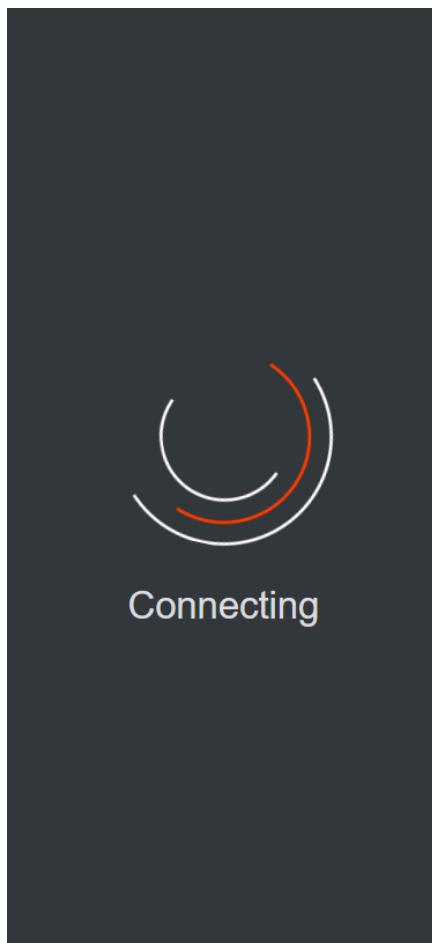
En obrir l'aplicació es crea un navegador Chrome sense el sistema de finestres, de manera que tots els menús estan amagats, i és redirigit a la pàgina, que està allotjada en un servidor extern. D'aquesta manera, tampoc s'ha d'actualitzar l'aplicació, ja que sempre es veu la versió actual.

El frontend està format per varis “pantalles” o capes superposades d'elements.

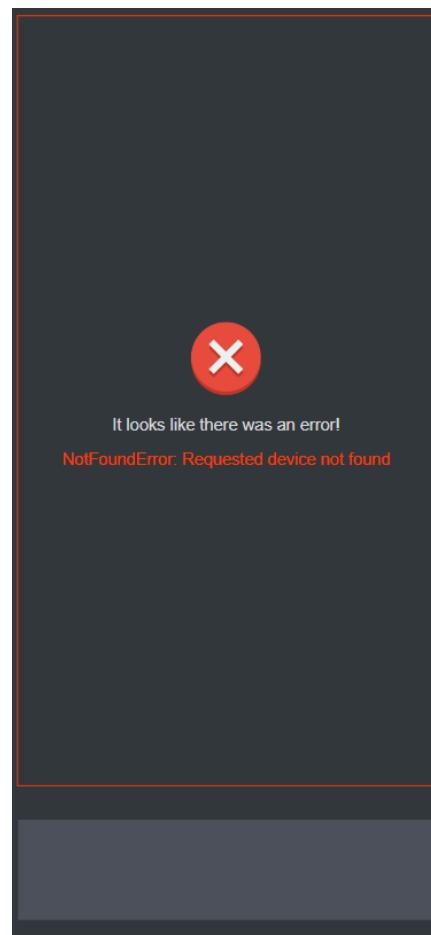
Aquestes són:

1. **Pantalla de càrrega que ocupa tota la pantalla:** Es mostra mentre es connecta al servidor. Normalment dura ~1 segon.
2. **Pantalla d'error:** Es mostra quan no s'ha trobat una càmera o no s'ha pogut accedir a ella.
3. **Pantalla de càrrega parcial:** Situada dins de la caixa de la vista prèvia de la càmera, és visible quan s'està inicialitzant la xarxa neuronal.
4. **Vista prèvia de la càmera:** Es mostra quan s'han processat totes les pantalles de càrrega. Llavors comença la detecció.

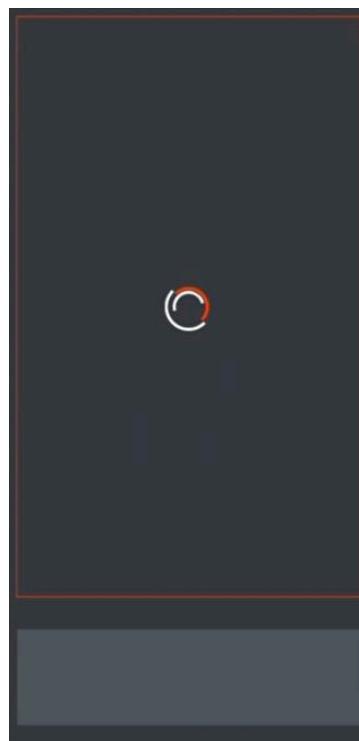
A continuació hi ha imatges de totes les pantalles.



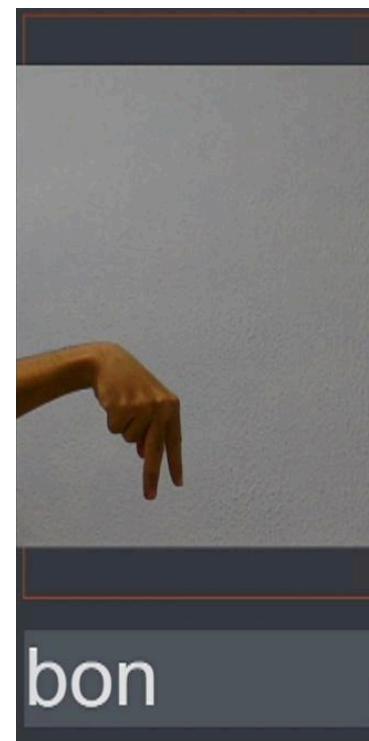
Primera pantalla de càrrega



Pantalla d'error



Segona pantalla de càrrega



Vista prèvia de la càmera

7.2 BACKEND

El backend (la part de l'aplicació que no pot veure l'usuari) es va allotjar en un servidor extern perquè fos accessible des de qualsevol part del món.

El servidor Linux té tres aplicacions sempre obertes:

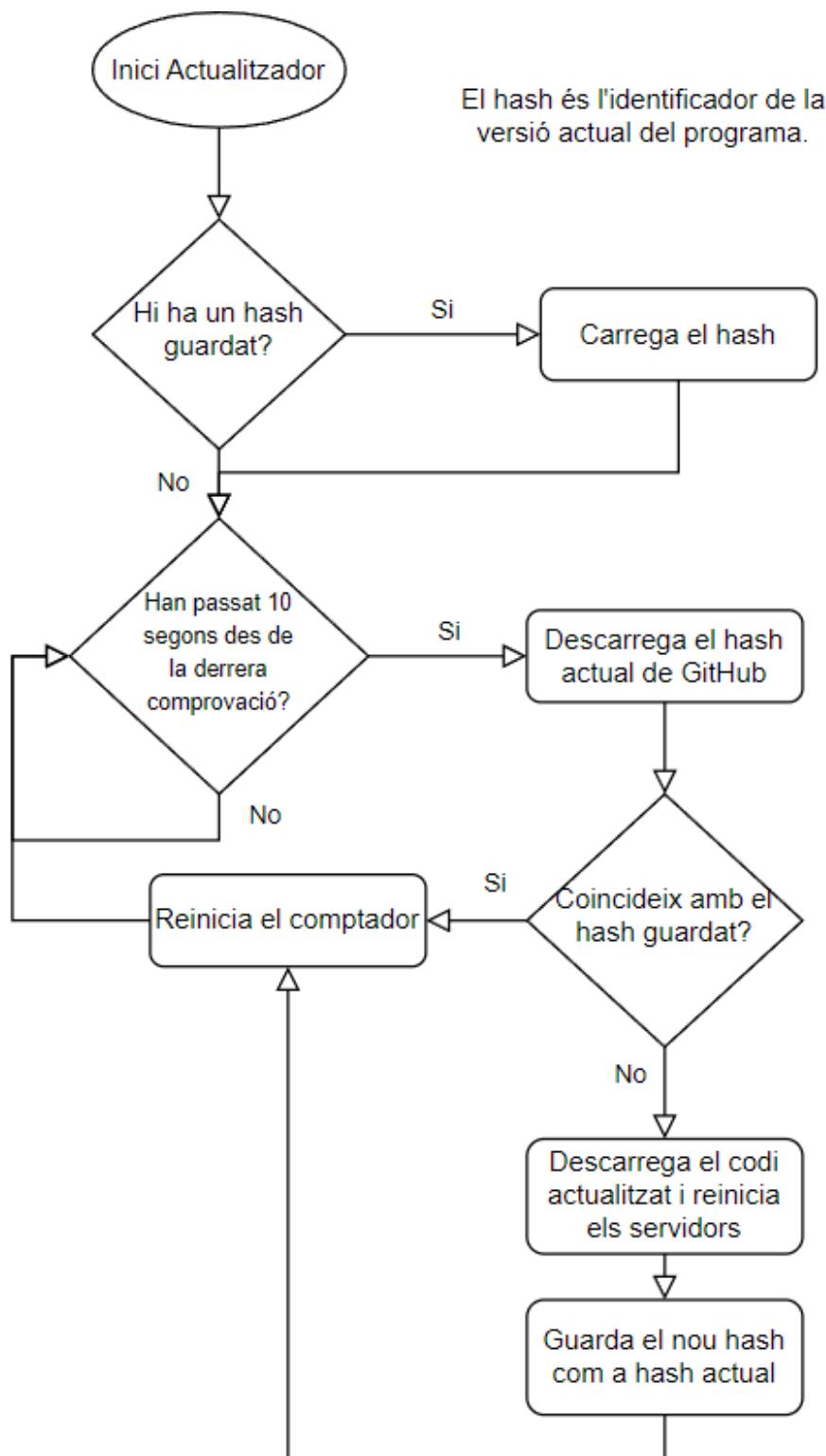
1. **L'actualitzador:** Descarrega actualitzacions i reinicia les altres dues aplicacions quan hi ha un canvi en el codi.
2. **El servidor web:** Allotja la web i tots els seus arxius, incloent-hi les llibreries de detecció.
3. **El servidor de classificació:** S'encarrega de classificar les mans que rep i retornar els resultats.

L'ACTUALIZADOR

L'actualitzador és un programa necessari per poder treballar ràpidament des d'un ordinador i que els canvis es transfereixin al servidor. Sense aquest, els canvis s'haurien de pujar manualment cada vegada, una solució bastant poc eficaç.

Funciona utilitzant la pàgina web [GitHub](#), en la que es pot pujar el codi d'una aplicació públicament i la gent el pot descarregar. Aprofitant aquest sistema, es pot pujar el codi sencer del projecte i l'actualitzador el pot descarregar.

Cada 10 segons el programa mira si hi ha hagut algun canvi en el codi de GitHub, si és diferent, el descarrega i reinicia els dos servidors. Aquest és el seu diagrama de flux:



Les altres dues aplicacions s'executen en pantalles separades en segon pla.

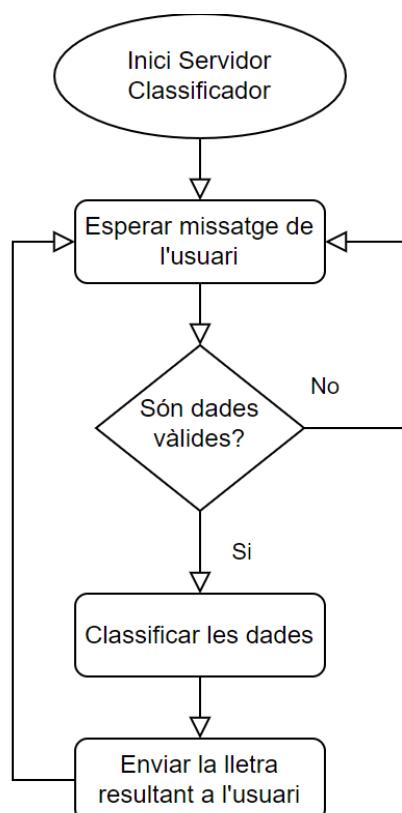
SERVIDOR WEB

L'únic propòsit del servidor web és enviar a l'usuari els arxius necessaris perquè es mostri la web i s'executi la detecció. Aquest retorna arxius d'HTML, CSS i JavaScript que formen la web i les tasques que fan que funcioni. Els arxius de JavaScript s'encarreguen de fer els canvis de pantalles de càrrega i detectar les mans amb la xarxa neuronal.

Tot segueix [l'esquema de la planificació](#).

SERVIDOR CLASSIFICADOR

El servidor classificador està a l'espera de qualsevol missatge d'un client. Quan el rep, classifica les dades enviades i retorna la lletra que s'ha de mostrar en pantalla.



Aquest també té una funció de gravació. Si s'inicia el servidor amb el paràmetre “rec” es gravaran les pròximes 100 captures de la mà a la base de dades. Això permet afegir nous signes fàcilment i només es pot fer si tens accés directe al programa.

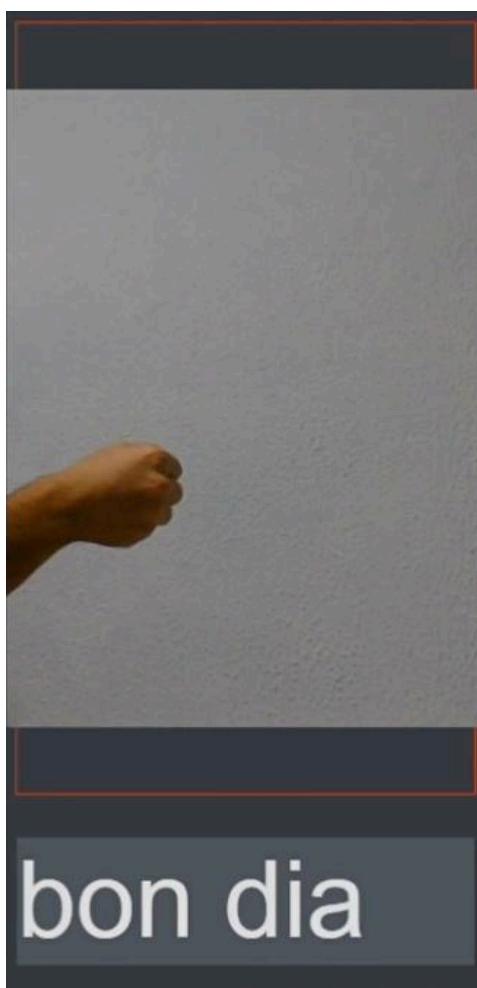
8. CONCLUSIONS

S'han assolit tots els objectius plantejats a la introducció. S'ha creat un sistema per detectar les mans i classificar les dades obtingudes, s'ha posat a prova i integrat en una aplicació.

Encara que no té molta utilitat en l'estat actual, aquest sistema podria ser millorat perquè es poguessin detectar signes en moviment o fins i tot dues mans. Però el projecte compleix el propòsit de demostració tècnica.

En conclusió, s'ha desenvolupat del projecte exitosament.

L'aplicació final consisteix en les 4 pantalles. Es pot fer clic en aquest enllaç per veure un vídeo del funcionament.



Oració sencera a l'aplicació

El seu rendiment arriba a més de 100 deteccions per segon, però com que són innecessàries tantes mostres, s'ha limitat a 10. Es llança i es carrega en menys de 20 segons en un dispositiu mòbil i detecta tots els signes possibles de l'alfabet dactilològic.

9. ANNEX

El codi de l'aplicació està pujat a GitHub, es pot fer clic en [aquest enllaç](#) per veure'l.

10. FONTS DE CONSULTA

La webgrafia i bibliografia consultades són:

Informació sobre el llenguatge de signes:

CNSE. Lengua de Signos. Sense data

<<https://www.cnse.es/index.php/lengua-de-signos>> [20/07/2022]

Generalitat de Catalunya. Llengua de signes catalana. Sense data

<https://llengua.gencat.cat/ca/llengua_signes_catalana/> [20/07/2022]

Barberà, Gemma. The meaning of space in Catalan Sign Language (LSC). 2012

<<https://www.tdx.cat/bitstream/handle/10803/81074/tgba.pdf>> [01/08/2022]

Anònim. Gramática de la lengua de signos. 2014

<<http://lenguadesignosnee.blogspot.com/p/221-gramatica-de-la-lengua-de-signos.html>> [01/08/2022]

Anònim. 70.000 personas usan la lengua de signos en España. 14 de juny de 2020

<<https://accedes.es/70-000-personas-usan-la-lengua-de-signos-en-espana/>> [04/08/2022]

Anònim. La historia del lenguaje de signos en españa. Sense data

<https://www.tenyus.com/productos-especificos/blog/la-historia-del-lenguaje-de-signos-en-espana_236_7_473_0_1_in.html> [06/08/2022]

Informació sobre la intel·ligència artificial i xarxes neuronals:

Anònim. *Redes neuronales artificiales e inteligencia artificial.* 6 de maig de 2021

<<https://blog.ticjob.es/redes-neuronales-artificiales-e-inteligencia-artificial/>>

[06/08/2022]

Mehta, Anukrati. *A Comprehensive Guide to Types of Neural Networks.* Sense data

<<https://www.digitalvidya.com/blog/types-of-neural-networks/>> [13/08/2022]

IBM Cloud Education. *Machine Learning.* 15 de juliol de 2020

<<https://www.ibm.com/cloud/learn/machine-learning>> [13/08/2022]

Informació sobre el teorema i classificador de Bayes:

Brownlee, Jason. *A Gentle Introduction to Bayes Theorem for Machine Learning.* 4

d'octubre de 2019

<<https://machinelearningmastery.com/bayes-theorem-for-machine-learning/>>

[21/08/2022]

Brownlee, Jason. *How to Develop a Naive Bayes Classifier from Scratch in Python.*

7 d'octubre de 2019

<<https://machinelearningmastery.com/classification-as-conditional-probability-and-the-naive-bayes-algorithm/>> [03/09/2022]

Brownlee, Jason. *Naive Bayes Classifier From Scratch in Python.* 18 d'octubre de

2019 <<https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/>>

[03/09/2022]